INFN
Istituto Nazionale
di Fisica Nucleare
Laboratori Nazionali di Frascati

SuperB

Feedback Operator Interface

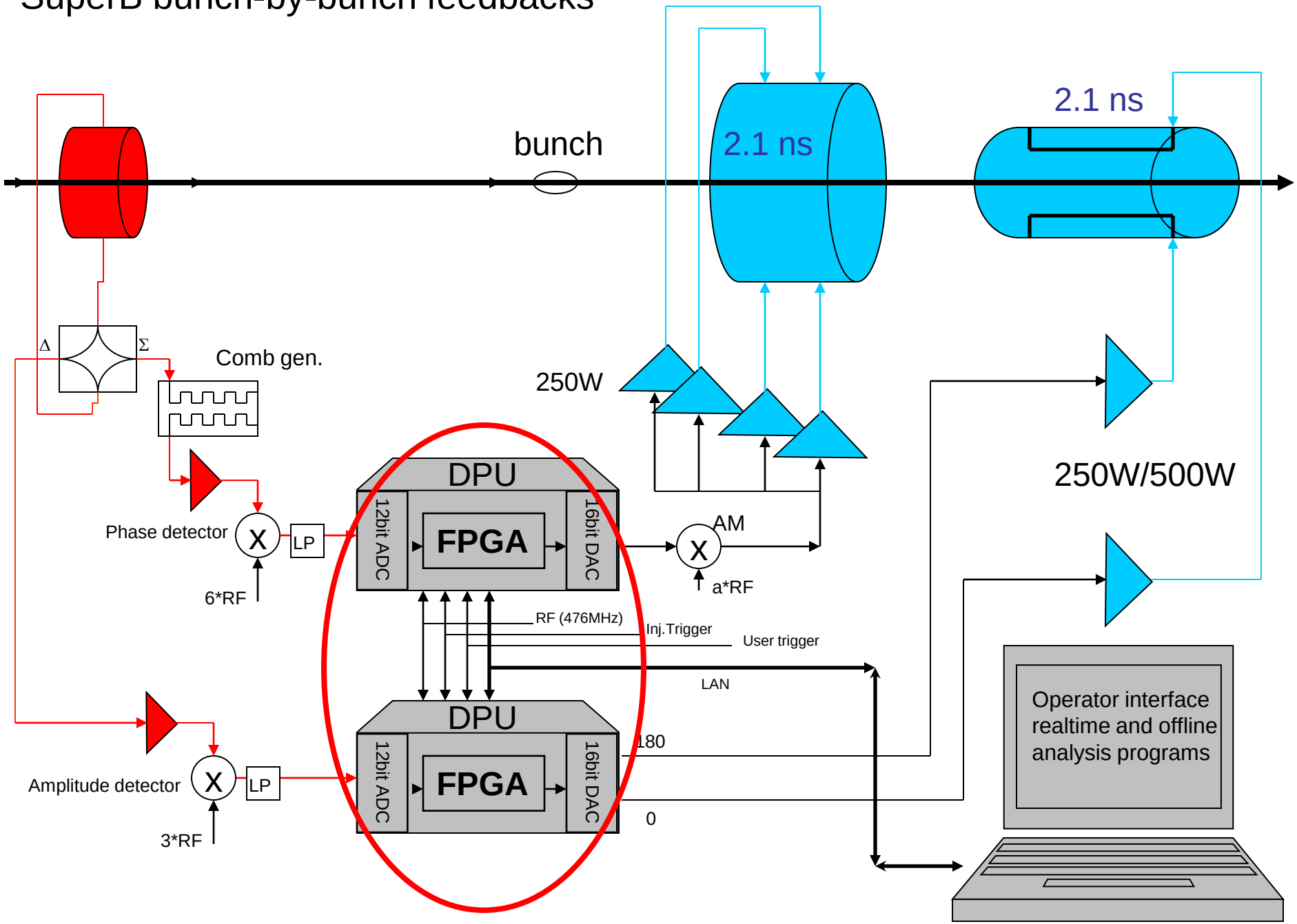
Alessandro Drago

XIV SuperB General Meeting - LNF, Sep/27-Oct/01/2010

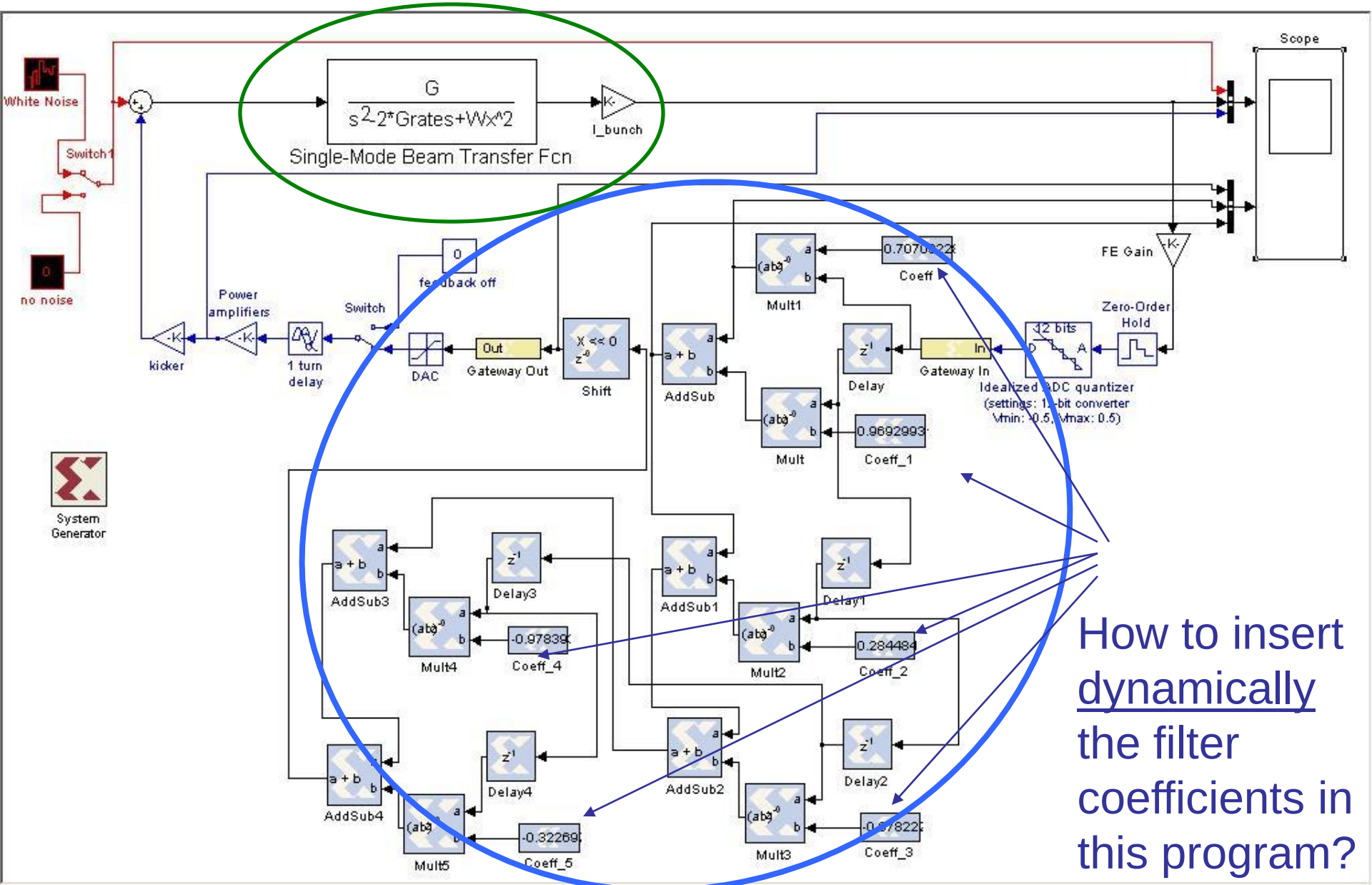
Introduction

- This talk is focused on the DPU (digital processing unit) that is the most important part of IP and Longitudinal/Transverse bunch-by-bunch feedbacks and on how to implement the remote operator interface
- Two different DPU systems are currently under study: the iGp12 by Dimtel, Inc. (an upgraded version of SLAC/KEK iGp system with 12bit ADC, 12bit DAC and Xilinx Virtex-5 FPGA inside) and a second, alternative design that is going to be in-house programmable, more compact and with newer parts
- Two iGp12 units have been ordered to Dimtel and they have just arrived at LNF– Tests with DAFNE beams are planned in the next weeks – DPU source code: NO
- The second system is mainly designed as IP feedback and therefore it requests even more sensitivity and dynamic range than the first one. On the other hand, it needs to be completely programmable to allow different working scheme and therefore it can also be implemented a classic bunch-by-bunch transverse or longitudinal feedback
- Main features: 14-bit ADC, 16-bit DAC, Virtex-7 FPGA, no pc inside for remote interfacing (evaluation in progress) - DPU source code: YES
- In the following the digital architecture and the operator interface for this second feedback system that is currently in R&D phase will be discussed

SuperB bunch-by-bunch feedbacks



Beam+feedback model generating FPGA code



How to insert dynamically the filter coefficients in this program?

3 ways to implement the operator i/f

A. PC + Linux + IOC (EPICS) with USB cable to feedback/FPGA module (like the iGp approach)

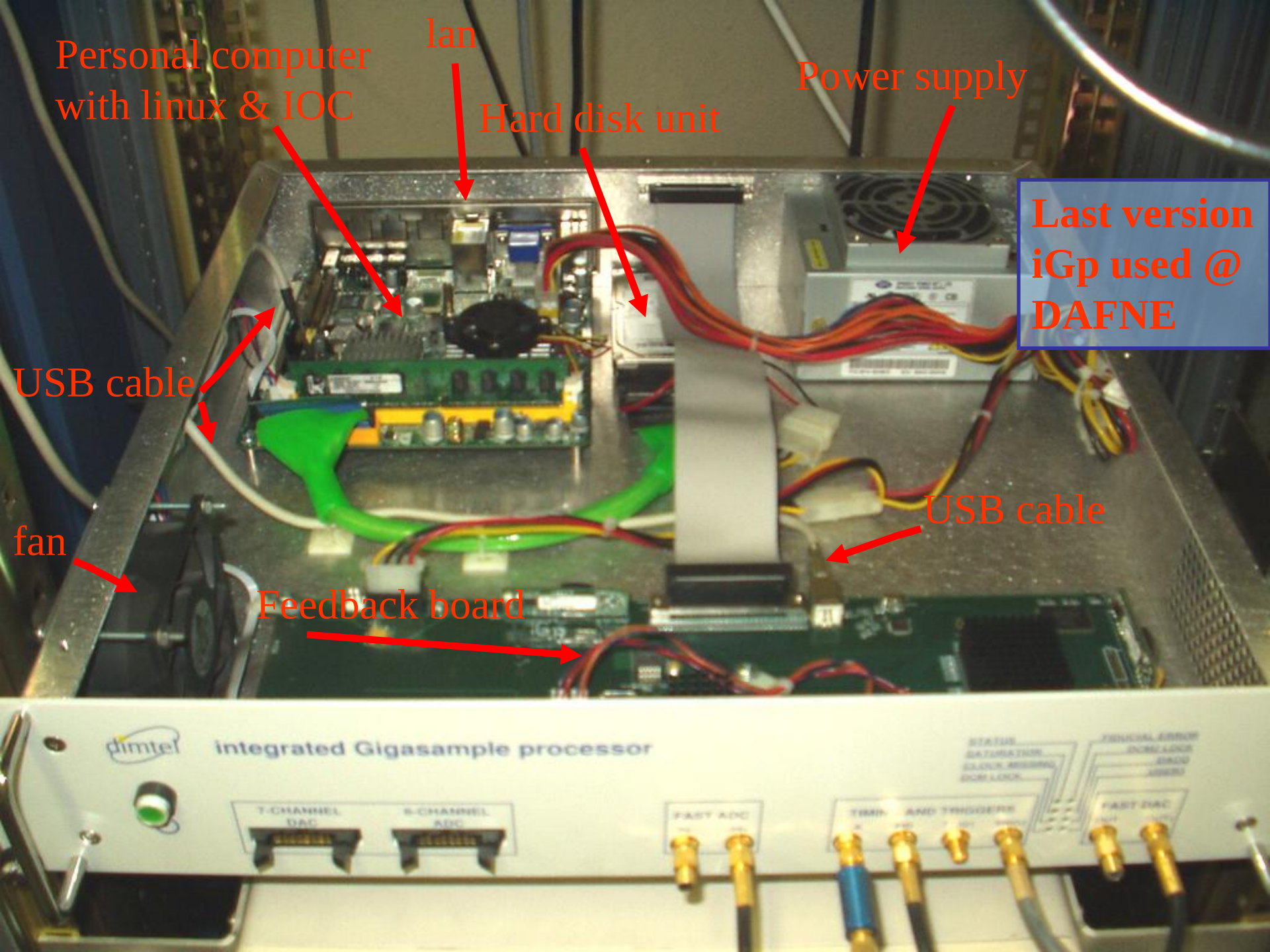
B. Embedded system (using ARM microcontroller) + linux (maybe Android by Google) + IOC (EPICS) software with a fast parallel bus to connect feedback (FPGA) module, in such a way similarly to the Bunch-by-bunch Libera feedback system by Instrumentation Technology

C. Embedded microcontroller (Microblaze or another one if available) designed inside the FPGA itself and directly connected by LAN (ethernet and WiFi) to remote operator interface and by dual-port ram to the feedback processing hardware

The first way is what we have now for iGp.
The second way is a sort of compromise.
In the next I will discuss briefly on the first 2 approaches,
but I will focus mainly on the third approach.
We will see that the choice is not neutral
(i.e. it impacts deeply on the system design)

Approach A.

PC + Linux + IOC (EPICS) with USB cable to feedback/FPGA module (like the iGp approach)


Personal computer
with linux & IOC
lan
Hard disk unit
Power supply
Last version
iGp used @
DAFNE
USB cable
USB cable
fan
Feedback board
dimtel
integrated Gigasample processor


dimtel
iGp12
8-CHANNEL DAC
8-CHANNEL ADC
FAST ADC
IN-
IN+
TIMING AND TRIGGERS
K
FID
TRIG1
TRIG2
STATUS
SATURATION
CLOCK MISSING
DCM LOCK
FIDUCIAL ERROR
DACO
USER1
USER2
FAST DAC
OUT+
OUT-

- iGp12, as the previous versions, is based on a pc connected to feedback board by USB interface

iGp12-120F Signal Processor

TECHNICAL USER MANUAL

Author:
Dmitry TEYTELMAN

Revision:
2.0

August 12, 2010

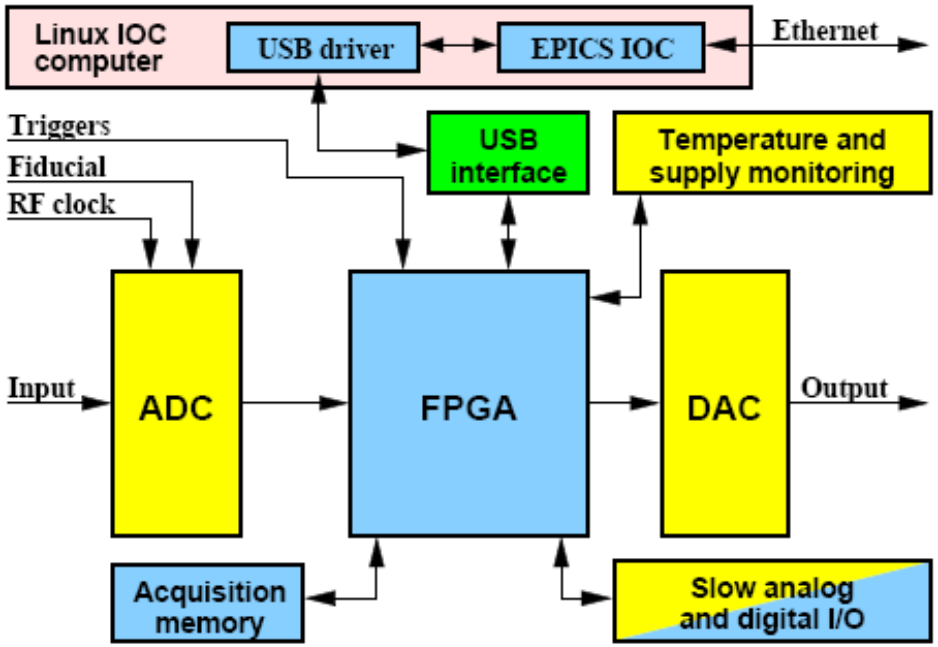
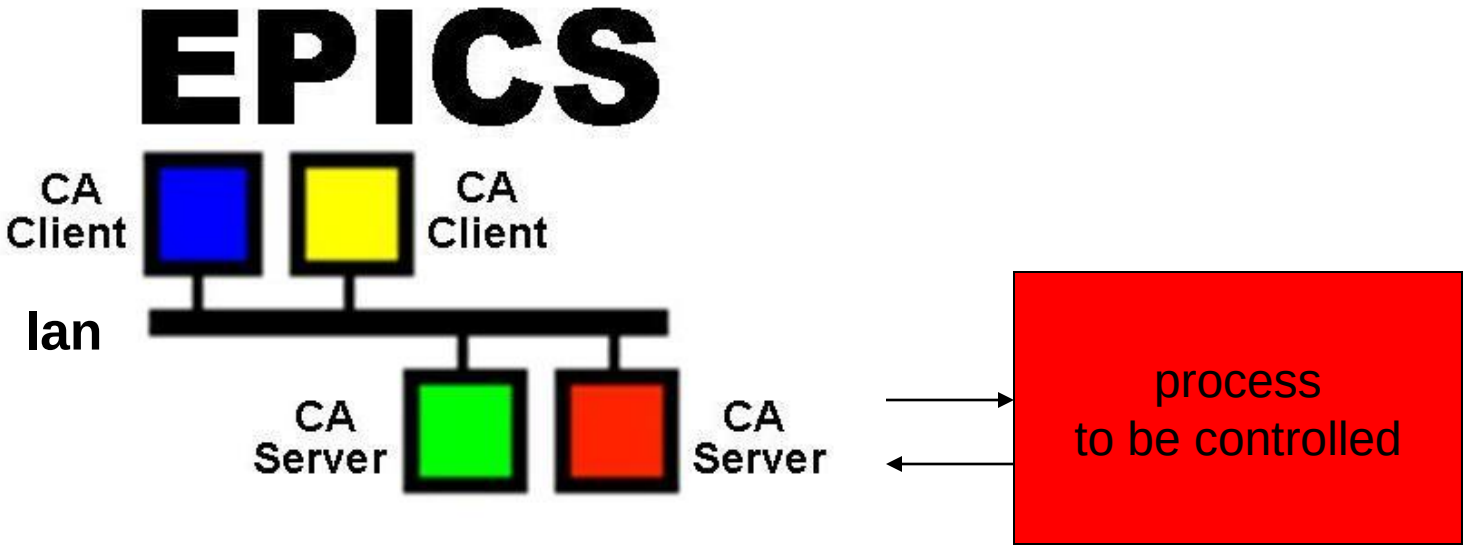


Figure 1: iGp12-120F block diagram

Operator interface and remote control: (based on EPICS)

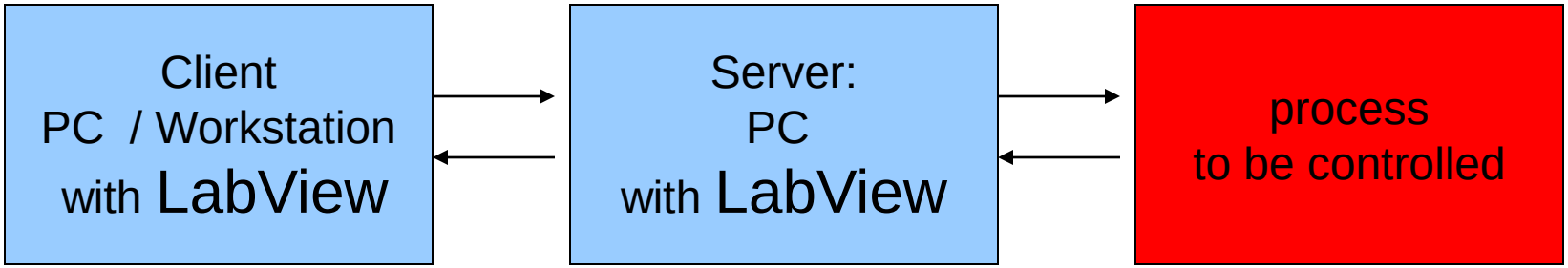
Network-based "client/server" model (hence the EPICS logo)



For EPICS, *client* and *server* speak of their Channel Access role

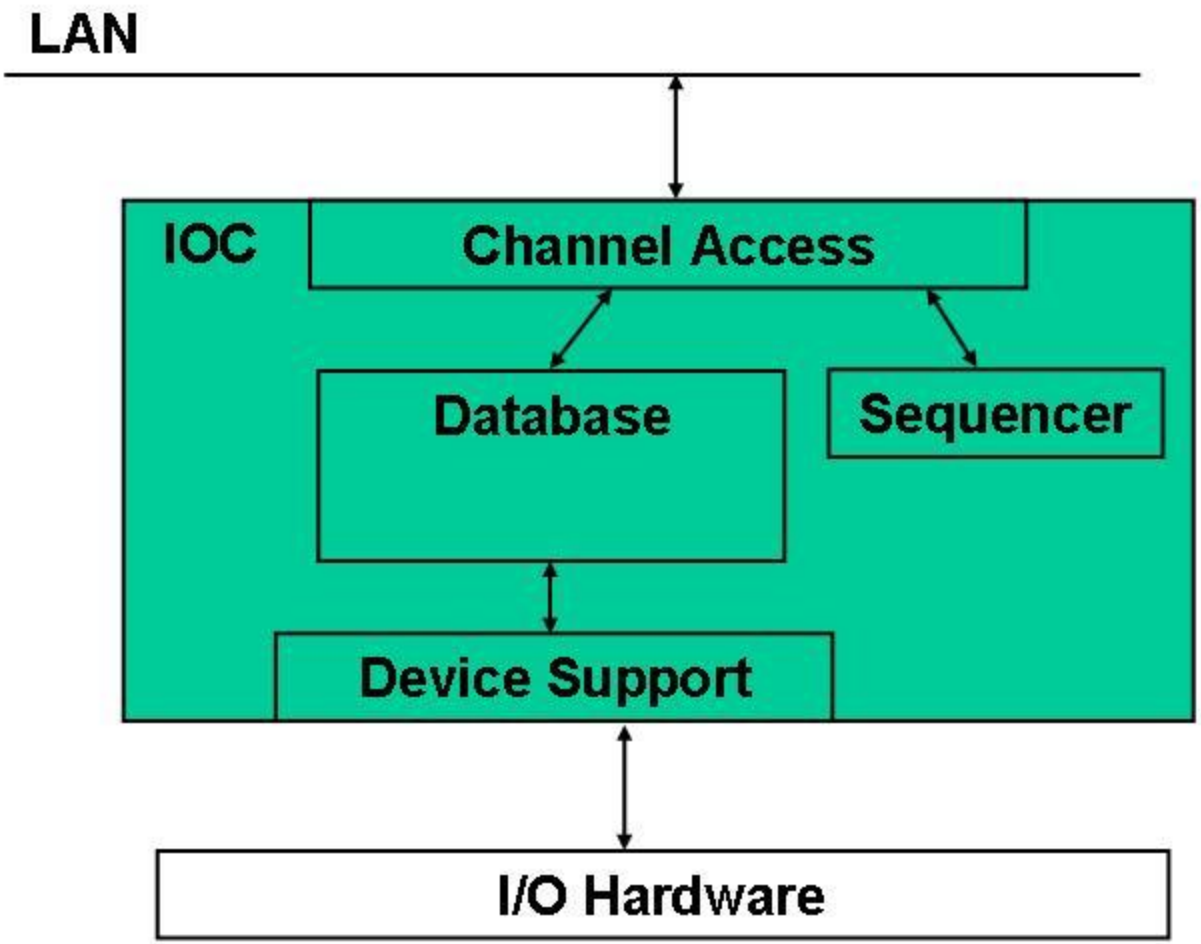
- *i.e. Channel Access Client & Channel Access Server*

A similar scheme could be based on LabView as in DAFNE control system



Inside an IOC *(Input/Output Controller)*

The major software components of an IOC (IOC Core)



Network Protocols

- **Channel Access uses two Network Protocols, UDP and TCP**
- **UDP (User Datagram Protocol)**
  - One way, unreliable
  - Send out packets, no guarantee they reach their destination
  - Can be broadcast or directed (unicasts)
    - *Broadcasts: To all IP addresses, e.g. 123.45.6.255*
    - *Unicasts: To a specific IP address, e.g. 123.45.6.100*
  - Broadcasts may not leave subnets for security reasons
- **TCP (Transmission Control Protocol)**
  - Two way, reliable, persistent
  - Socket at each end
  - Acknowledgements, timeouts, retransmissions, etc. guarantee reliability

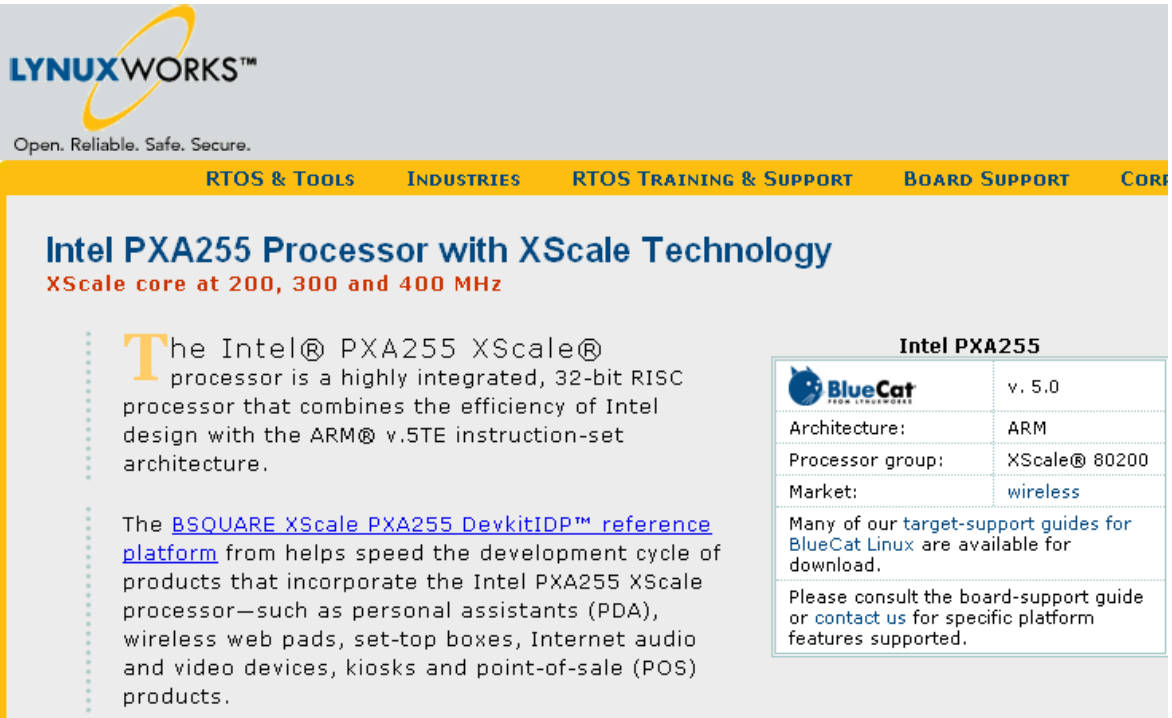
Approach B.

Embedded system (using ARM microcontroller) + linux (that could be Android by Google) + IOC (EPICS) software with a fast parallel bus to connect feedback / FPGA module, in such a way similarly to Bunch-by-bunch Libera feedback system by Instrumentation Technology

Bunch-by-bunch Libera feedback by Instruments Technologies

- Bunch-by-bunch Libera feedback uses an SBC (single board computer) based on INTEL PXA 255-400 MHz a microcontroller based on ARM instruction set
- Single-board computer is used for running the Linux operating system and performing tasks like housekeeping, running application software and interfacing with the control system.
- The SBC is connected to the FPGA (feedback) module by a bus
- Other features:
- Analog input signal acquired by four 14-bit ADC but split in four ways (each with sampling frequency = RF/4)
- Analog bandwidth: 400MHz (poor!)
- FPGA: Virtex-II Pro
- 14-bit DAC
- Flexible digital signal processor code

- PXA is a microprocessor that makes part of XScale, a microprocessor core, Intel's and Marvell's implementation of the ARMv5 architecture, that consists of several distinct families: IXP, IXC, IOP, CE and PXA
- The XScale architecture is based on the ARMv5TE **without** the floating point instructions.
- All the generations of XScale are 32-bit ARMv5TE processors



Good points for the ARM microcontroller are that the EPICS IOC software already exists and many linux distributions are available

On the market many ARM based boards are sold. In this market Google is pushing for Android, a powerful linux operative system distribution with many application under R&D

ARM Boards with Ethernet

These ARM boards feature an Ethernet interface.

Displaying **1** to **20** (of **78** products)

Result Pages: **1** 2 3 4 [Next >>]

	Product Name	Model	Price+
	Philips LPC2129 (ARM) Ethernet Board, USB, CAN	LPC-E2129	US$74.95
	ARM926EJ-S CPU Board, 4MB Flash, 32MB RAM, Ethernet, Linux	Eddy-CPU-V2.0	US$79.00
	ARM926EJ-S Module, Ethernet, USB, MMC, Lemonix	Eddy-CPU-V2.1	US$89.00
	NXP LPC2138 Development Board, Ethernet-programmable	ARMweb	US$99.00
	LPC2294 (ARM7) Board, Ethernet, CAN, RS232, SD/MMC, 1MB SRAM	LPC-L2294	US$99.95
	Philips LPC2124 (ARM) Ethernet Board, USB	LPC-E2124	US$102.95
	NXP LPC2378 (ARM) Prototype Development Board, USB, 2x CAN	LPC-P2378	US$105.95
	ARM9 Linux based 100Mbps Ethernet to RS232 or RS422/458 Module	Eddy-S1-DB9-V2.0	US$115.00
	ARM9 Linux based 100Mbps Ethernet to RS232/RS422/458 pin header	Eddy-S1-PIN-V2.0	US$115.00
	LPC2138 (ARM7) Board: Ethernet, 2x RS232, Keypad, LCD	MINI-MAX/ARM-E	US$119.00

Approach C.

Embedded microcontroller (Microblaze or another one if available) designed inside the FPGA itself and directly connected by LAN (ethernet / WiFi) to remote operator interface and by dual-port ram to the feedback processing hardware

Which advantages in c) design ?

- Only one development tool for all the digital system ➔ more simple code management ➔ more flexibility and easily for next features or corrections
- A more compact system can be useful in the crowded IP zone: it can be put very close to the signals from pickups and to kicker having less propagation delay
- Less power consumption and, accordingly, less heating
- In case of multiple transverse, longitudinal or IP feedback systems, it will be easier to find space in the SuperB tunnel
- Speaking in general, it’s a more modern and advanced design

MicroBlaze embedded processor has 32bit CPU with 4 Gbytes addressing capability and 32 general purpose registers

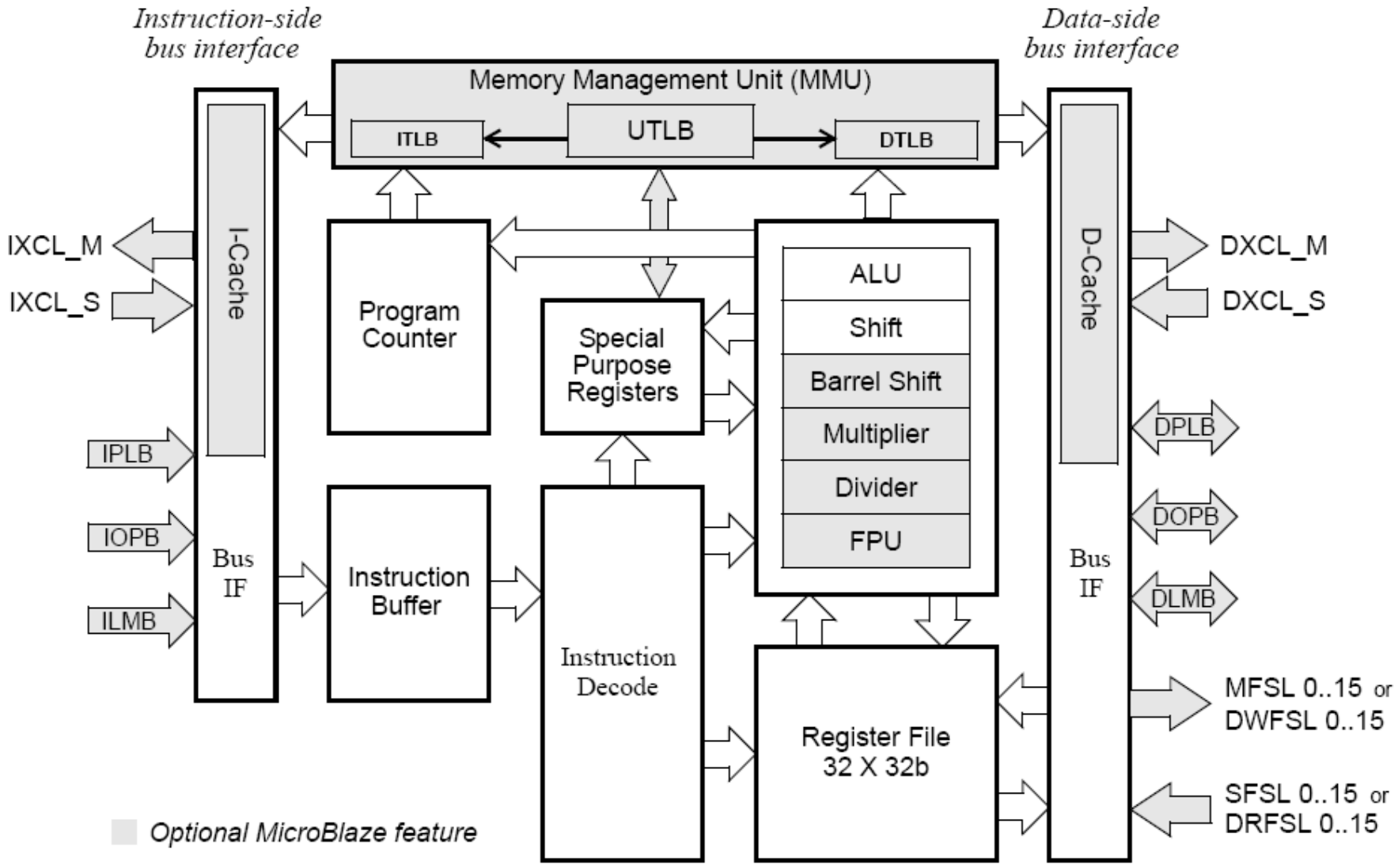


Figure 1-1: **MicroBlaze Core Block Diagram**

Microblaze main features

- In terms of its instruction-set architecture MicroBlaze has a RISC architecture.
- With few exceptions, the MicroBlaze can issue a new instruction every cycle, maintaining single-cycle throughput under most circumstances.
- The MicroBlaze has a versatile interconnect system to support a variety of embedded applications.
- MicroBlaze's primary I/O bus, the CoreConnect PLB bus, is a traditional system-memory mapped transaction bus with master/slave capability.
- Many aspects of the MicroBlaze can be user configured: cache size, pipeline depth (3-stage or 5-stage), embedded peripherals, memory management unit, and bus-interfaces can be customized.
- The performance-optimized version expands the execution-pipeline to 5-stages, allowing top speeds of 210 MHz on Virtex-5 fpga family.
- Also, key processor instructions which are rarely used can be selectively added/removed (i.e. multiply, divide, and floating-point operations) – the FPU is available.
- With the memory management unit, MicroBlaze is capable of hosting operating systems requiring hardware-based paging and protection, such as the Linux kernel. Otherwise it is limited to operating systems with a simplified protection and virtual memory-model: e.g. FreeRTOS or Linux without MMU support.

MicroBlaze uses LwIP routines for implements lan communication features. LwIP is a “lightweight” protocol and it has been tested this month successfully in the LNF local area network (though with some limitations versus a PC)

lwIP

From Wikipedia, the free encyclopedia

lwIP (*lightweight IP*) is a widely used open source TCP/IP stack designed for embedded systems. lwIP was originally developed by Adam Dunkels at the Swedish Institute of Computer Science and is now developed and maintained by a world wide network of developers led by Kieran Mansley.

lwIP is used by many manufacturers of embedded systems. Examples include Altera (in the Nios II operating system), Analog Devices (for the Blackfin DSP chip), Xilinx and Honeywell (for some of their FAA certified avionics systems).

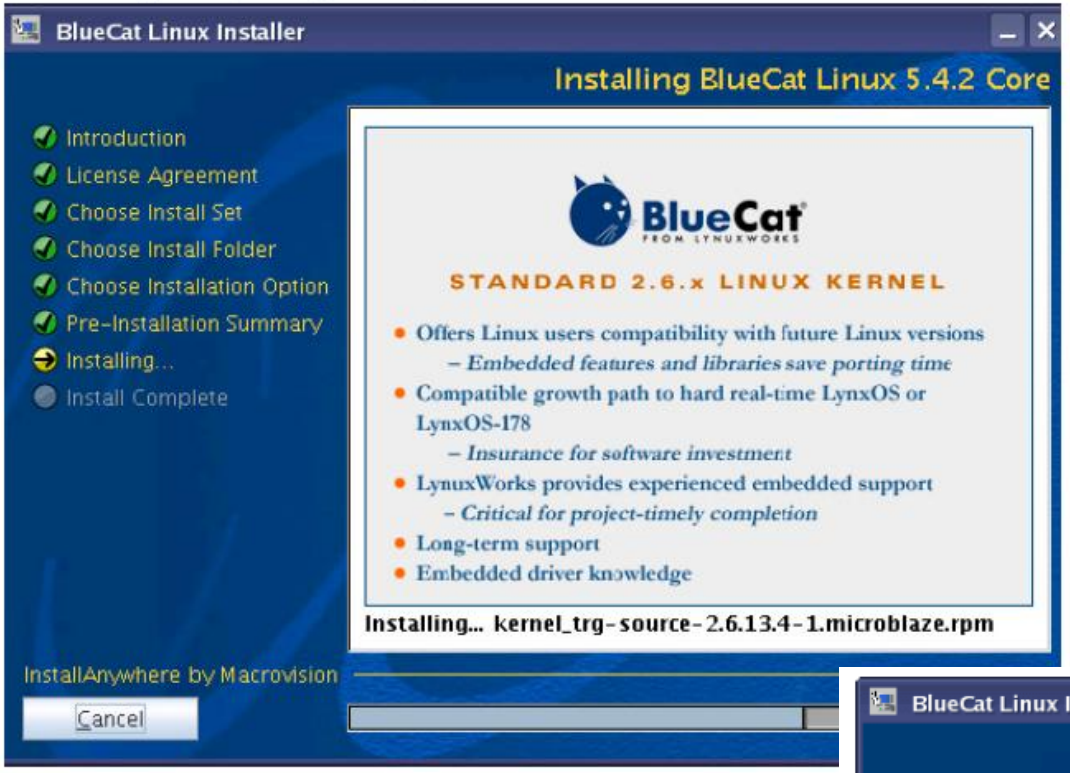
The focus of the lwIP TCP/IP implementation is to reduce resource usage while still having a full scale TCP. This makes lwIP suitable for use in embedded systems with tens of kilobytes of free RAM and room for around 40 kilobytes of code ROM.

lwIP

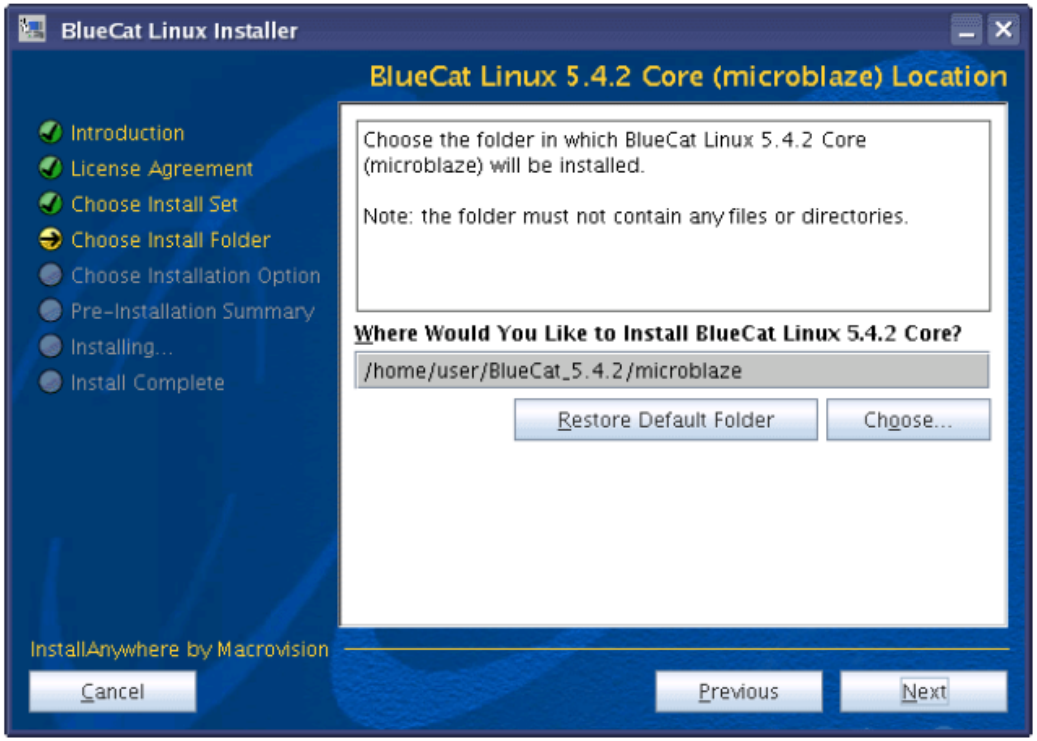
Original author(s)	Adam Dunkels
Developer(s)	lwIP developers group
Development status	active
Written in	C
Operating system	multiple
Platform	embedded systems
Type	IP stack
License	Modified BSD license
Website	http://savannah.nongnu.org/projects/lwip/

lwIP features [edit]

- IP (Internet Protocol) including packet forwarding over multiple network interfaces
- ICMP (Internet Control Message Protocol) for network maintenance and debugging ← tested by ping
- IGMP (Internet Group Management Protocol) for multicast traffic management
- UDP (User Datagram Protocol) including experimental UDP-lite extensions ← → used by EPICS
- TCP (Transmission Control Protocol) with congestion control, RTT estimation and fast recovery/fast retransmit ← tested by http
- Specialized raw/native API for enhanced performance
- Optional Berkeley-like socket API
- DNS (Domain names resolver)
- SNMP (Simple Network Management Protocol)
- DHCP (Dynamic Host Configuration Protocol) ← Not implemented in the test code, because not necessary
- AUTOIP / Link-local address (for IPv4, conforms with RFC 3927)
- PPP (Point-to-Point Protocol)
- ARP (Address Resolution Protocol) for Ethernet



Microblaze can use BlueCat Linux (with standard 2.6.x Linux kernel)
even though other operative systems are available



But:

1) an EPICS distribution package for Microblaze still doesn't exist !!!
2) linux + EPICS inside Microblaze could be too slow
3) And moreover Microblaze has only the C compiler, not the LabView compiler

In the first tests the feedback operator interface has been based on **http protocol and web browser**: an advantage is that the device can be easily connected also to smart phones and tablets

LAN

LAN

WiFi

Xilinx Virtex–7 FPGA

Software

Dual port ram

Gateware

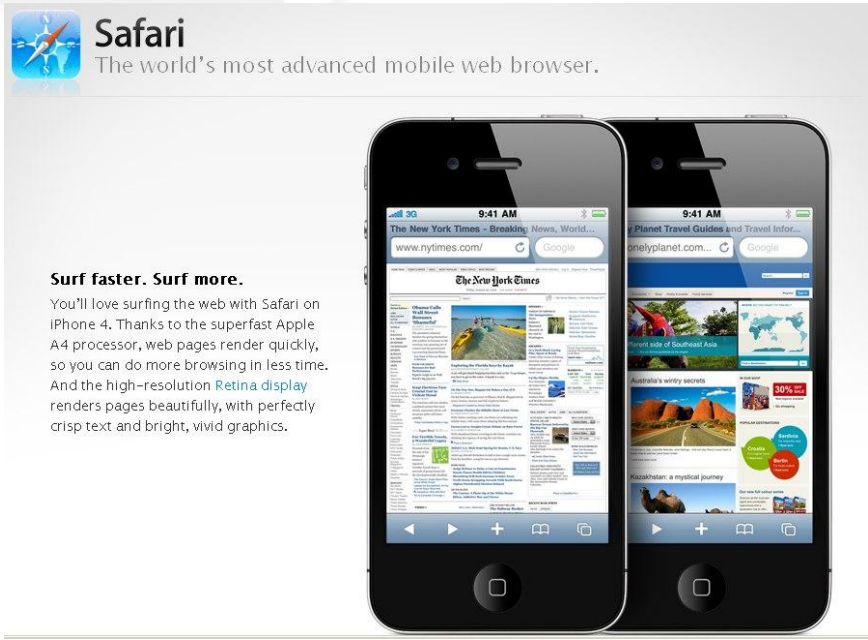
[feedback digital hardware]

Feedback board

Another interesting feature is that a single chip FPGA can implement more than 1 Microblaze (up to 8 can be tested by JTAG interface)

Looking to the next 5-10 years, tablets and smart phones will be largely used as remote terminals in control systems

iPad and iPhone have growing popularity and lowering prices



Web browsers (and WiFi) are included in tablets and smart phones and, more important, the web technology has continuous and very fast evolution

Microblaze – feedback interface

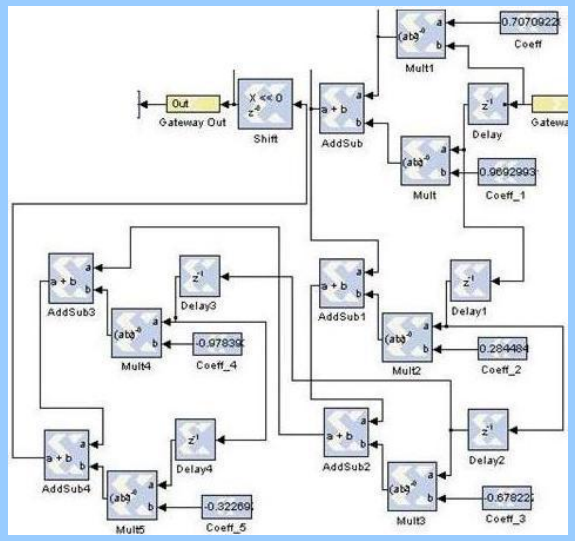
Xilinx Virtex – 7 FPGA

Software

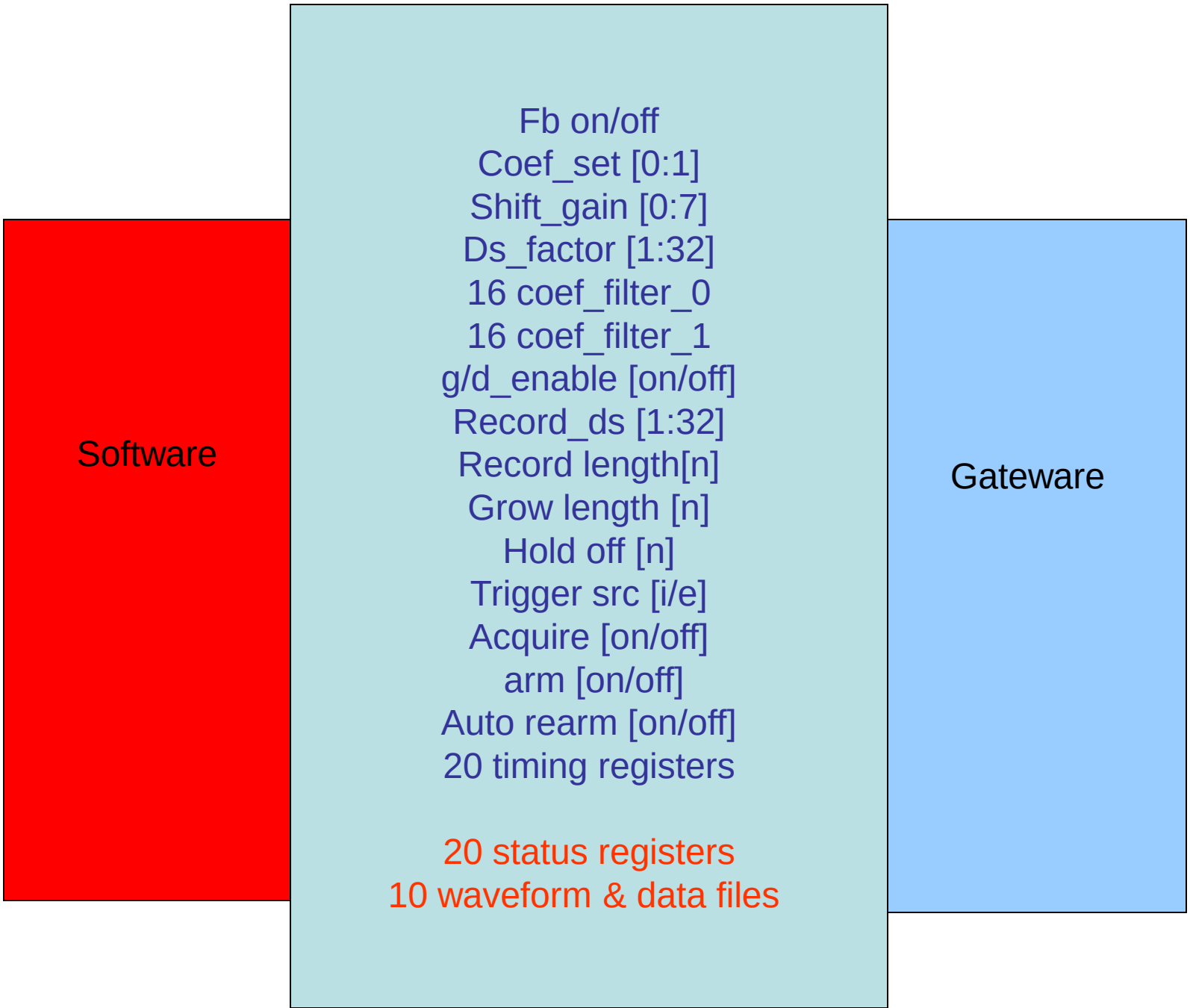
Microblaze controller (with flash memory) and Uart and ethernet connections managed by “C” programs
&
LWIP - 1.3.2

Dual port ram

Gateware [feedback digital hardware]



Microblaze – feedback dual port ram data base



From EPICS tech – talk

Subject: **HTML Device Driver**
From: David Dudley <ddudley@bnl.gov>
To: EPICS tech-talk <tech-talk@aps.anl.gov>
Date: Wed, 28 Jul 2010 11:33:50 -0400
*Title: **HTML Device Driver***

Here's a new one for you (Got Ralph and Michael going about how to do it, anyway ;-)

Almost all of the PLC equipment we're going to be using on NSLS/2 is connected on Ethernet.

It is common practice in the communications modules now, to integrate an embedded WEB server into the device. Both the Siemens and Rockwell hardware we're going to use have that capability, as well as most of the new generation of instrumentation.

Does anyone have knowledge of an EPICS device driver that is capable of reading and interpreting a WEB page or XML page? Seems that this would be the most universal way to read data from all this differing equipment, if such a driver was available.

David Dudley

Replies:

- Re: HTML Device Driver *Pete Jemian*
- Re: HTML Device Driver *J. Lewis Muir*
- Re: HTML Device Driver *Rod Nussbaumer*
- Re: HTML Device Driver *emmanuel_mayssat*

An EPICS interface could be also designed as top level layer above http layer:
A recent post discuss how to interface web instruments from EPICS

From EPICS tech – talk /2

Subject: **Re: HTML Device Driver**
From: "J. Lewis Muir" <jlmuir@anl.gov>
To: EPICS Tech-Talk <tech-talk@aps.anl.gov>
Date: Wed, 28 Jul 2010 11:14:45 -0500

Hi, David.

Are you asking about a generic EPICS device driver for communicating with devices via HTTP and capable of interpreting HTML or XML, or are you asking about existing drivers that do this for a particular device?

I've written two drivers that communicate w/ the device via HTTP and then extract the needed information from an HTML response:

http://www.imca.aps.anl.gov/~jlmuir/sw/dli-epcr.html
http://www.imca.aps.anl.gov/~jlmuir/sw/websensor-em01b.html

These drivers use asyn but have hacks to deal w/ the connection getting closed after each request which asyn did not handle well before asyn 4-13. According to the asyn 4-13 release notes, new behavior has been added to handle this by specifying "http" as the protocol in drvAsynIPPortConfigure. I haven't tried it yet.

Certainly something generic could be written to make it easy to construct the HTTP request, handle any authentication, and handle the response. This might provide something like the httplib module in Python.

I think it would get more difficult if you wanted to go beyond that. A simple method for interpreting the response is to use regular expression matching to extract the desired information. One would then typically want to convert that text into some more appropriate value (e.g. a double) in the driver.

A more powerful approach would be to actually provide an HTML or XML parser allowing access to the response via SAX or a DOM. I'm not sure how this would work for devices that return invalid HTML or XML responses. I would bet that many devices do not actually respond with valid HTML or XML.

Lewis

Pocket PC Applications

- **EPICS has been ported to the Pocket PC**
- **Two EPICS applications are available**
  - ProbeCE
  - BeamDisplay

- **There are better ways to access EPICS with a Pocket PC**
  - E.g. Citrix Metaframe
  - Can give access to all the EPICS WIN32 Extensions
  - Covered in the presentation on Remote Access

Hardware status update and tests done at LNF

TEXAS INSTRUMENTS
www.ti.com

ADS5474

SLAS525–JULY 2007

14-Bit, 400-MSPS Analog-to-Digital Converter

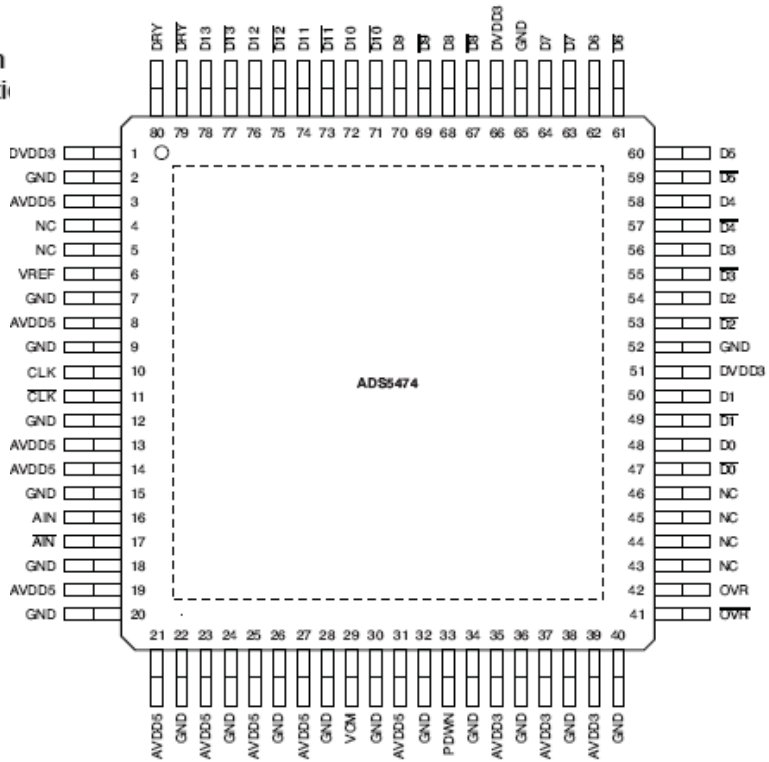
FEATURES

- 400-MSPS Sample Rate
- 14-Bit Resolution, 11.2-Bits ENOB
- 1.4-GHz Input Bandwidth
- SFDR = 80 dBc at 230 MHz and 400 MSPS
- SNR = 69.8 dBFS at 230 MHz and 400 MSPS
- 2.2 V_{PP} Differential Input Voltage
- LVDS-Compatible Outputs
- Total Power Dissipation: 2.5 W
- Power Down Mode: 50mW
- Offset Binary Output Format
- Output Data Transitions on the Rising and Falling Edges of a Half-Rate Output Clock
- On-Chip Analog Buffer, Track-and-Hold, and Reference Circuit
- TQFP-80 PowerPAD™ Package (14 mm × 14 mm footprint)
- Industrial Temperature Range: –40°C to +85°C
- Pin-Similar/Compatible with 12-, 13-, and 14-Bit Family: ADS5463 and ADS5440/ADS5444

APPLICATIONS

- Test and Measurement Instrumentation
- Software-Defined Radio
- Data Acquisition
- Power Amplifier Linearization
- Communication Instrumentati
- Radar

ADS5474 by Texas Instruments has been ordered

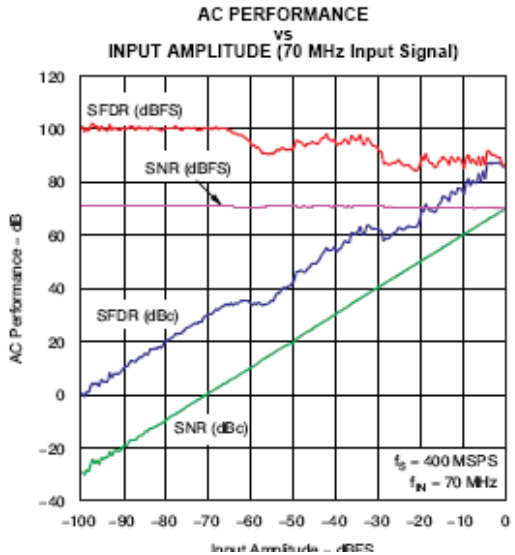


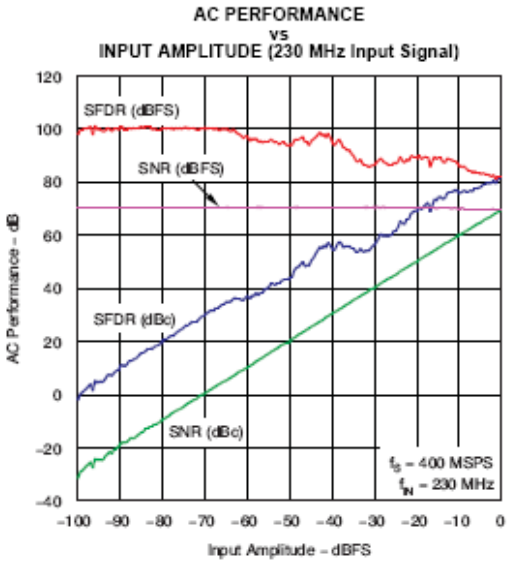
14bit Analog to Digital Converter
with
~70dB signal to noise ratio

It will be interesting to compare noise levels with 12 bits versus 14 bits conversion on real beam signals during DAFNE runs

TYPICAL CHARACTERISTICS (continued)

At T_A = +25°C, sampling rate = 400 MSPS, 50% clock duty cycle, 3-V_{PP} differential sinusoidal clock, analog input amplitude = –1 dBFS, AVDD5 = 5 V, AVDD3 = 3.3 V, and DVDD3 = 3.3 V, unless otherwise noted.





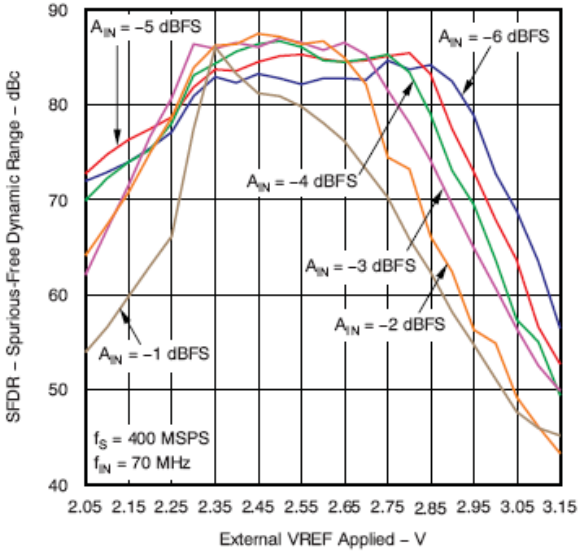


Figure 36. SFDR versus External VREF and A_{IN}

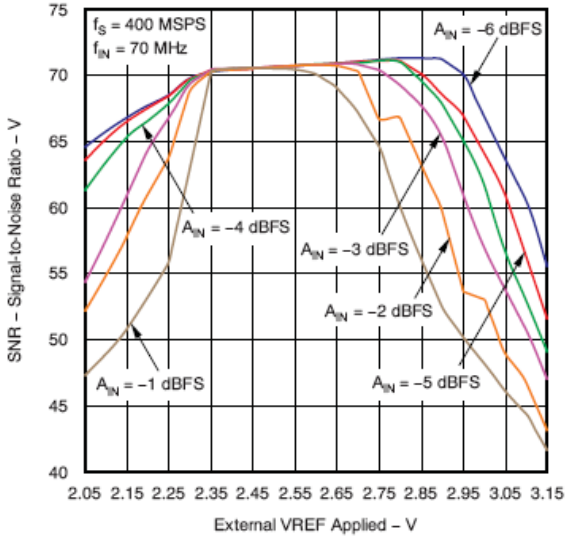
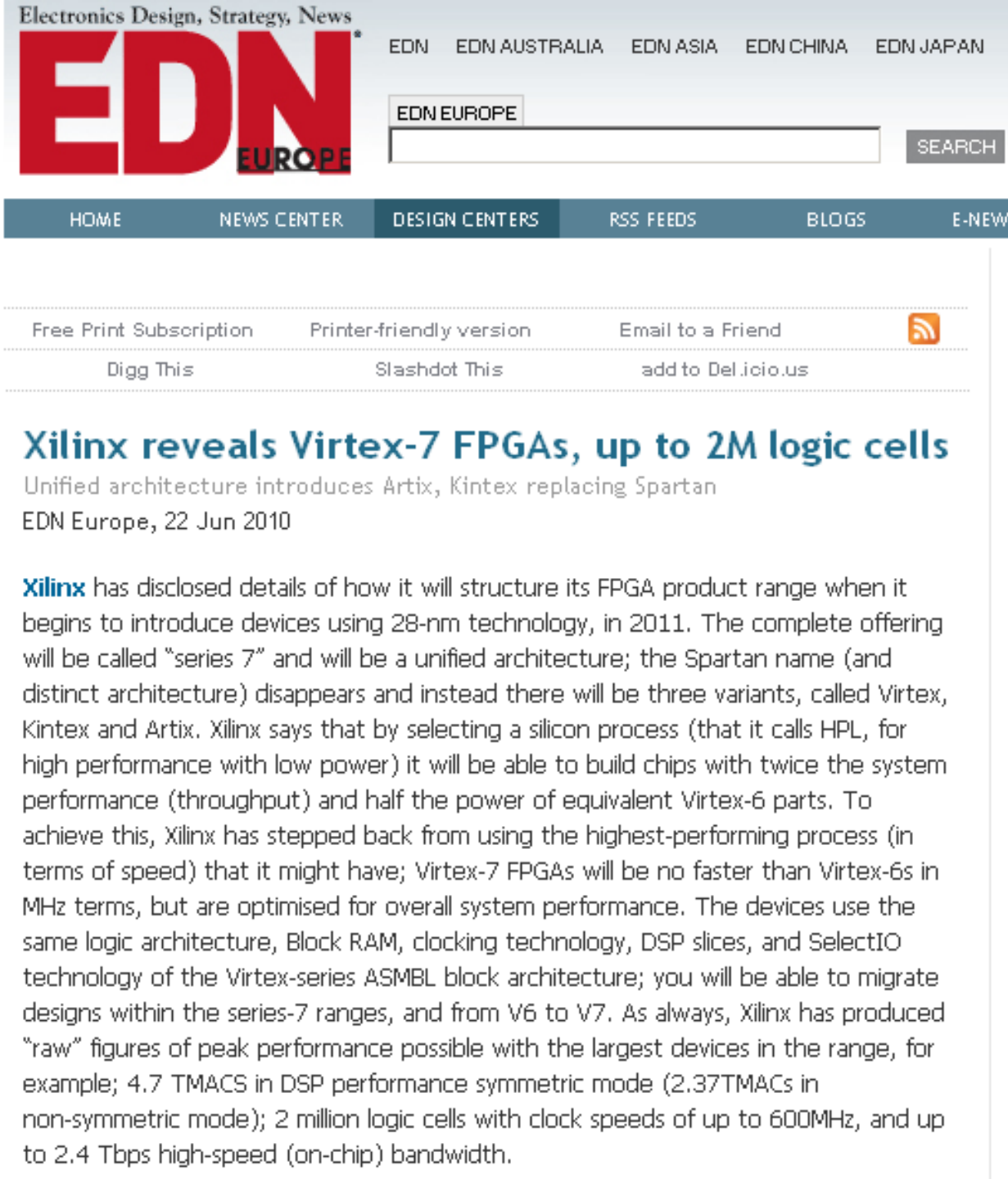


Figure 37. SNR versus External VREF and A_{IN}

- ADC very good performance
- Output signals in LVDL logic levels perfectly compatible with FPGA

Xilinx announces that Virtex-7 production will start in Feb / 2011

This announcement seems to discourage new designs based on series 6, because in series 7, though maximum speed remains the same, consumption halves and density doubles.
In addition, series 7 has followed very early series 6 that doesn't yet take off and has few applications.



Electronics Design, Strategy, News

EDN EUROPE

EDN | EDN AUSTRALIA | EDN ASIA | EDN CHINA | EDN JAPAN

EDN EUROPE

SEARCH

HOME | NEWS CENTER | DESIGN CENTERS | RSS FEEDS | BLOGS | E-NEW

Free Print Subscription | Printer-friendly version | Email to a Friend

Digg This | Slashdot This | add to Del.icio.us

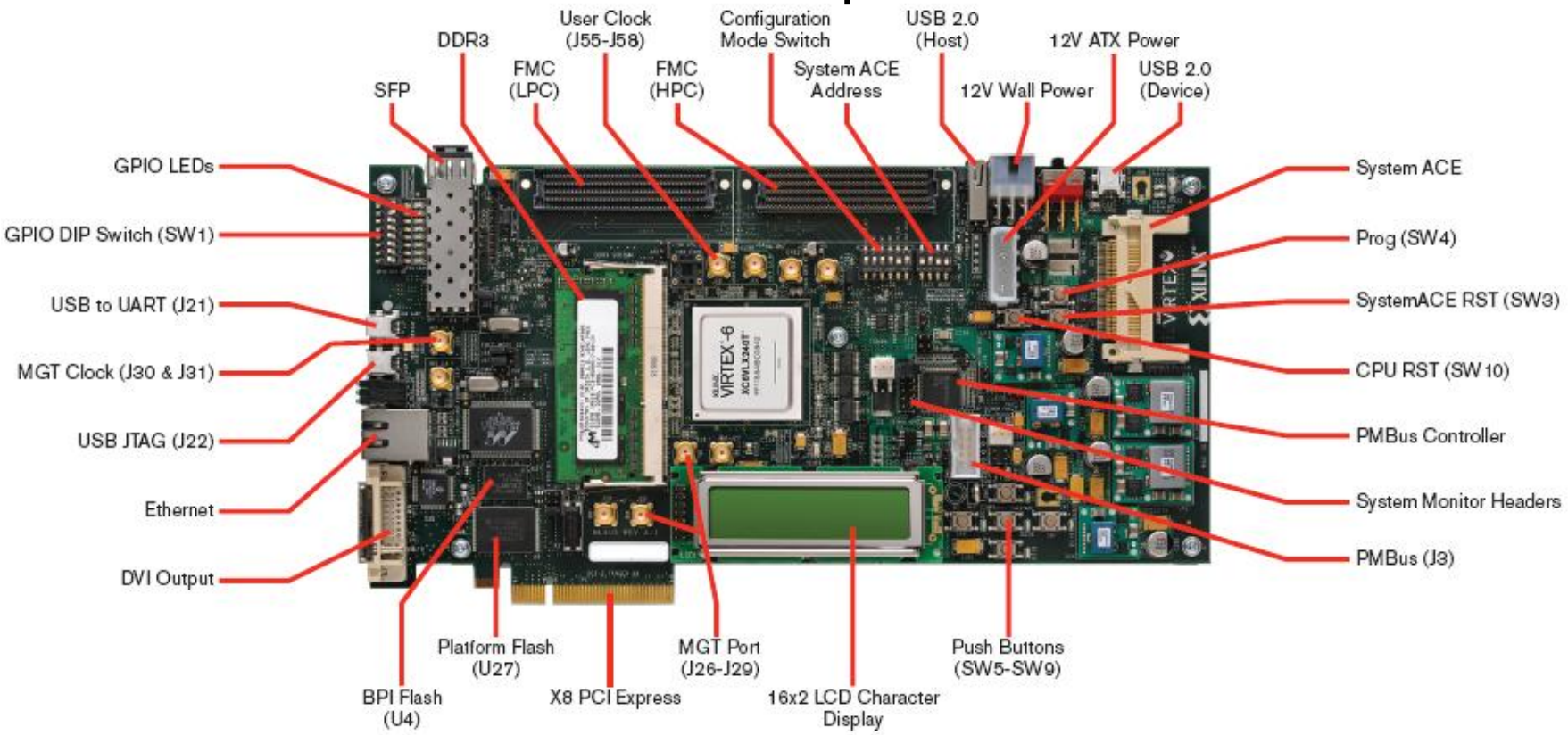
Xilinx reveals Virtex-7 FPGAs, up to 2M logic cells

Unified architecture introduces Artix, Kintex replacing Spartan

EDN Europe, 22 Jun 2010

Xilinx has disclosed details of how it will structure its FPGA product range when it begins to introduce devices using 28-nm technology, in 2011. The complete offering will be called "series 7" and will be a unified architecture; the Spartan name (and distinct architecture) disappears and instead there will be three variants, called Virtex, Kintex and Artix. Xilinx says that by selecting a silicon process (that it calls HPL, for high performance with low power) it will be able to build chips with twice the system performance (throughput) and half the power of equivalent Virtex-6 parts. To achieve this, Xilinx has stepped back from using the highest-performing process (in terms of speed) that it might have; Virtex-7 FPGAs will be no faster than Virtex-6s in MHz terms, but are optimised for overall system performance. The devices use the same logic architecture, Block RAM, clocking technology, DSP slices, and SelectIO technology of the Virtex-series ASMBL block architecture; you will be able to migrate designs within the series-7 ranges, and from V6 to V7. As always, Xilinx has produced "raw" figures of peak performance possible with the largest devices in the range, for example; 4.7 TMACS in DSP performance symmetric mode (2.37TMACs in non-symmetric mode); 2 million logic cells with clock speeds of up to 600MHz, and up to 2.4 Tbps high-speed (on-chip) bandwidth.

It's a fact that the Virtex-6 FPGA evaluation board, Xilinx ML605, ordered last April, still has not arrived



All the R&D work is currently done on Virtex-5 boards (ML506) that has similar performance being able to connect the FPGA board to the remote operator without the use of a personal computer as server (tests completed in September 2010 at LNF)

Operator Interface web-based: tested or in progress basic function list

- Action types:
  - Local action from operator to server database (tested)
  - Digital command from operator to FPGA (tested)
  - Digital status or alarm from FPGA to operator (tested)
  - Write value from operator to FPGA (tested)
  - Read value from FPGA to operator (tested)
  - Continuous read value from FPGA (in progress)
  - Read data block from FPGA to server (tested)
  - Download data file from server to operator (tested)
  - Write data block from server to FPGA (in progress)
  - Database memory after web browser closing (tested/updated in Sept/2010)

Feedback Operator Interface preliminary function list

- Turn on / off
- Filter bank choice: [0/1]
- Shift gain [0:7]
- Downsampling [1:32]
- New FIR filter coefficients entry:
  - gain [0 : 1]
  - phase [-360 : +360]
  - center frequency [0.00 : 1.00]
  - n_taps [1 : 16]
  - Raw coefficients entry [coef_0 : coef_15]
  - Filter coefficients plot
  - Filter magnitude plot
  - Filter phase plot

Tools from LightWeight IP (LwIP) library

- Web server for MicroBlaze
  - Simplified web server management with only two calls: get and post
- http get
  - It can be used to access to files stored in the server memory file system “memfs”
- http post
  - It writes and reads object status (registers, memory locations, led’s, switches, etc.)

Other tested tools for .html pages: the YUI Library

The YUI Library is a set of utilities and controls, written with JavaScript and CSS, for building richly interactive web applications using techniques such as DOM scripting, DHTML and AJAX.
YUI is available under a BSD license and is free for all the uses.

YUI 2 library calls included in the present tested code

- <script type="text/javascript" src="yui/yahoo.js"></script>
- The YAHOO object is the single global object used by YUI Library. It contains utility function for setting up namespaces, inheritance, and logging. YAHOO.util, YAHOO.widget, and YAHOO.example are namespaces created automatically for and used by the library.
- <script type="text/javascript" src="yui/dom.js"></script>
- Internal methods used to add style management functionality to DOM.
- <script type="text/javascript" src="yui/event.js"></script>
- The Event Utility provides utilities for managing DOM Events and tools for building event systems
- <script type="text/javascript" src="yui/conn.js"></script>
- The Connection Manager provides a simplified interface to the XMLHttpRequest object. It handles cross-browser instantiantion of XMLHttpRequest, negotiates the interactive states and server response, returning the results to a pre-defined callback you create.
- The Connection Manager singleton provides methods for creating and managing asynchronous transactions.
- <script type="text/javascript" src="yui/anim.js"></script>
- Base animation class that provides the interface for building animated effects.
- <script type="text/javascript" src="js/main.js"></script>
- main.js is not a YUI 2 call, it is a program written by the user

Conclusions

- An upgraded version of iGp feedback system (with 12bit ADC and 12bit DAC) has just arrived at LNF and will be tested with DAFNE beams in the next weeks
- As alternative (necessary for IP feedback) it seems possible to design a DPU (digital processing unit) with 14-bit ADC and 16-bit DAC and without a personal computer to interface the remote operator: this will make more compact the system and more flexible the design
- Preliminary tests seem to show the good feasibility of a control software based on web browser approach, even though many functions have to be still written & tested
- Nevertheless, EPICS developers are also evaluating how to interface web server remote devices using IOC
- R&D is in progress, of course the source code needs to be almost completely rewritten with respect to the first, very old, version of the iGp feedback system