Building software systems for Xilinx FPGAs

Assumed Hardware

- Virtex-4 FX12 FPGA
  - PowerPC 405 Processor
  - 10/100/1000 Ethernet Port
- 32M x 16 DDR Memory
- RS232 Port
- System ACE Interface and/or JTAG Port
- Compact Flash I/F and/or Flash Memory

Available Hardware

- Xilinx Development Kit ML403 or ML405
- Memec Virtex-4 FX12 Mini-Module
- Your own design! (base it on ML403!!)

Awaiting solutions using Virtex-5 FX

Performance

- Assume use of 300MHz clock for the PPC processor core
- Assume 64Mbyte DDR

- Compares with MVME2432 VME processor

- Ethernet (using TEMAC) gives real world TCP transfers at 12 Mbyte/sec with basic options (including DMA). Up to 32 Mbyte/sec with all options enabled (such as offloading checksum calculation) have been obtained at cost of using more FPGA resources.

Software Options

Consider software and firmware as equivalent

- VHDL with standalone C program in the PPC core
- VHDL with C code using proprietary OS in the PPC core
- VHDL with Linux OS in the PPC core

Proprietary OS

- U-Boot
  - Universal bootloader - feature overload!!
- Xilinx Kernel
  - Multi-threaded C application
  - lwIP Library (Light Weight IP)
    - supports BSD style sockets API
- Others

Linux

- Linux Version 2.6
- Requires Xilinx EDK 9.1.1
  - ❿Supports TEMAC for 1000BaseT
- Boot from platform flash or compact flash card
- Root filesystem on compact flash card or network server

Linux device drivers

- opb_uartlite for UART 16550
- plb_temac for 10/100/1000 Mbit ethernet support
- opb_iic for EEPROM access
- opb_sysace for Compact Flash access

- opb_gpio – good example base for “user” drivers

- Other drivers
  - plb_uart16550 – requires Xilinx IP core
  - plb_ethernet – 10/100 ethernet support
  - plb_tft
  - opb_ps2

Building Linux System

- Use Xilinx EDK to build hardware bitstream
  - Combines IP cores for required Xilinx specific devices
  - Plus any “user” devices
  - Plus “user” VHDL code
- Build the Linux kernel
- Create System ACE file which combines bitstream file and kernel file => .ace file

Data Acquisition

- Once the Linux OS has been loaded you can then build and load your acquisition application in much the same way as when using for example a VME processor board.
- Use “shared” memory (address space) to create a private “bus” between the PPC core and your VDHL code.
- You will need a Linux driver which gives access to this shared address space allowing communication between your acquisition application and your VHDL code within the FPGA.

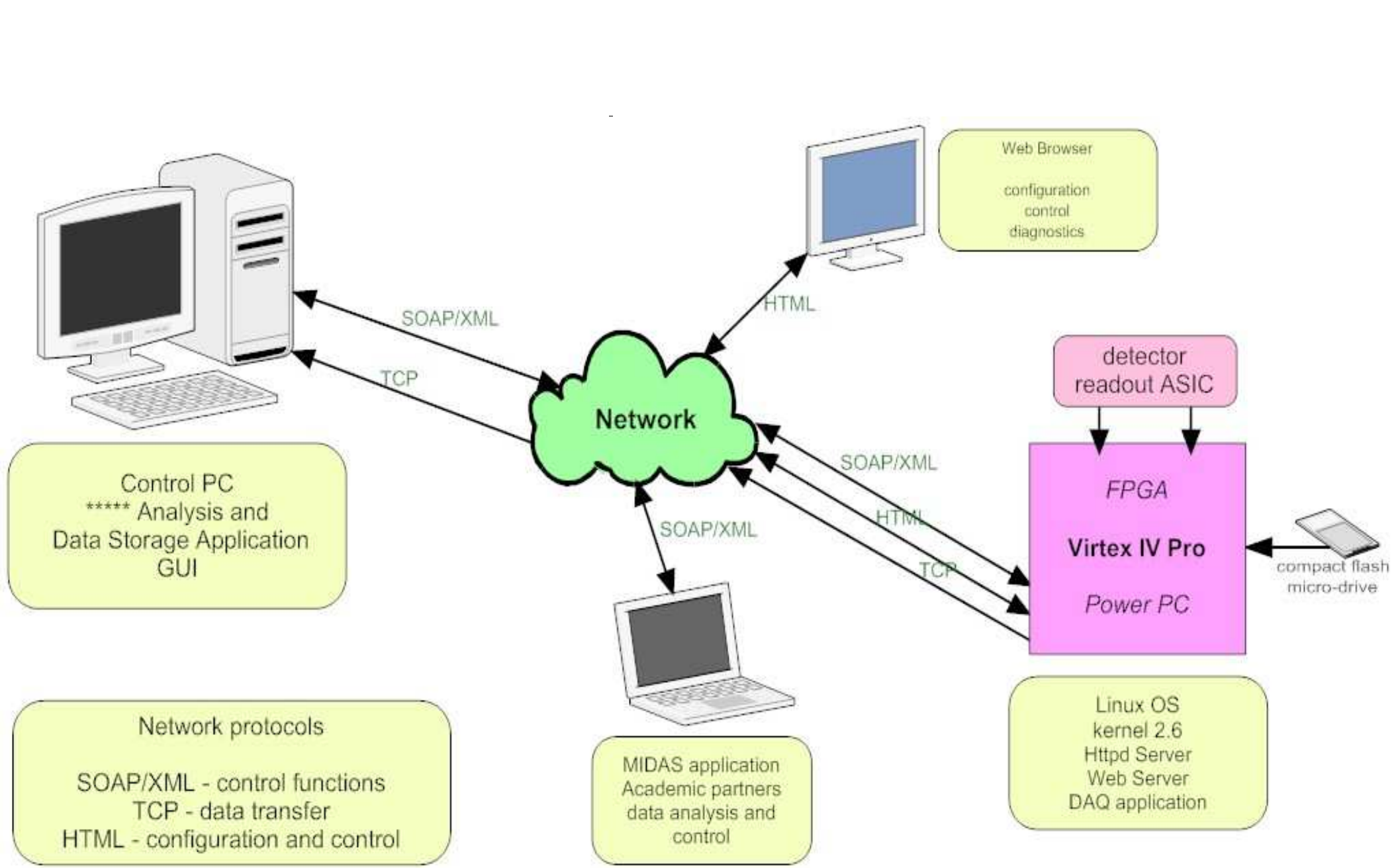
Advantages

- Once the embedded Linux system is operational all existing code from, for example, VME based applications can very quickly be ported to the Virtex-4 environment.
- All non hardware specific features of the application will be available
- This should then immediately give an acquisition application as reliable as the original VME application.
- Development of the application can use the GNU toolkit which is likely to be much quicker than any other solution
- All advantages based on Linux developments are likely to be available

Disadvantages

- You require the Xilinx software which defines the peripherals (uart and temac) in order to build the essential bitstream file.

- This may take some time after availability of hardware.



********** System Structure

FREEDAC Building Blocks

- need access to time reference source in order to time-stamp all data items
- application builds data stream to agreed format
- application sends time-stamped data items to event builder/data concentrator using agreed data transfer protocol