



The World Leader in PCI Silicon and Software Technology

Processor Independent I₂O Solutions Using Shell and Private Platform

Mark Easley

www.plxtech.com





Today's Agenda

- ◆ I₂O Shell
- ◆ I₂O Messaging
- ◆ Processor Independent Private Platform Solutions
- ◆ PCI Bridges and I/O Processors

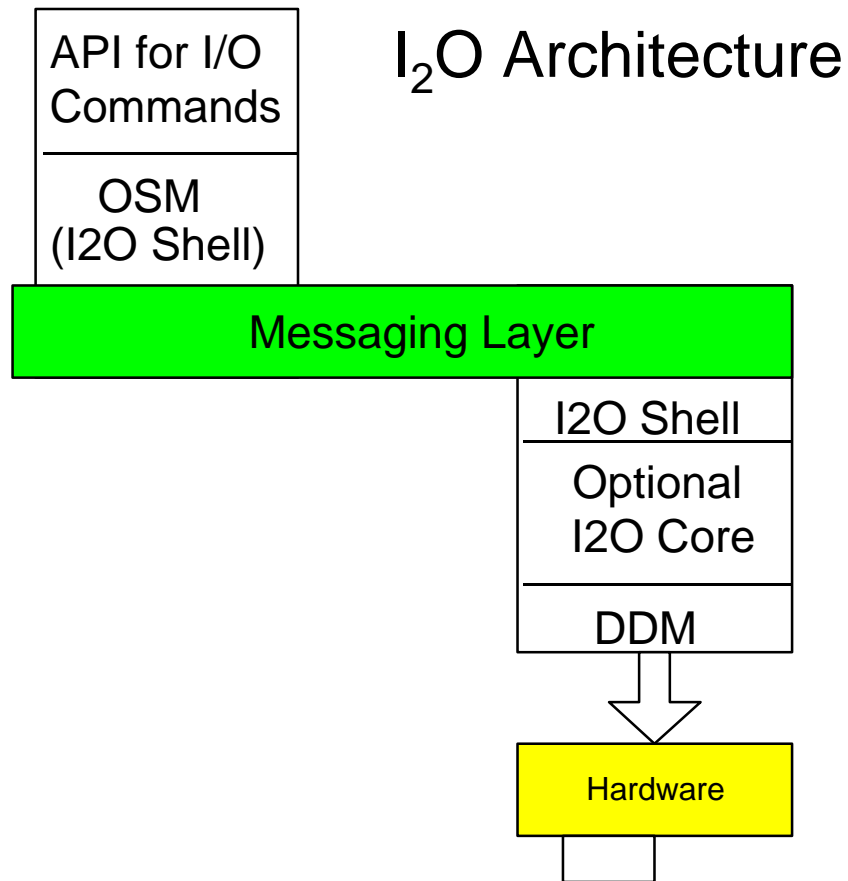
Feel free to ask questions at any time



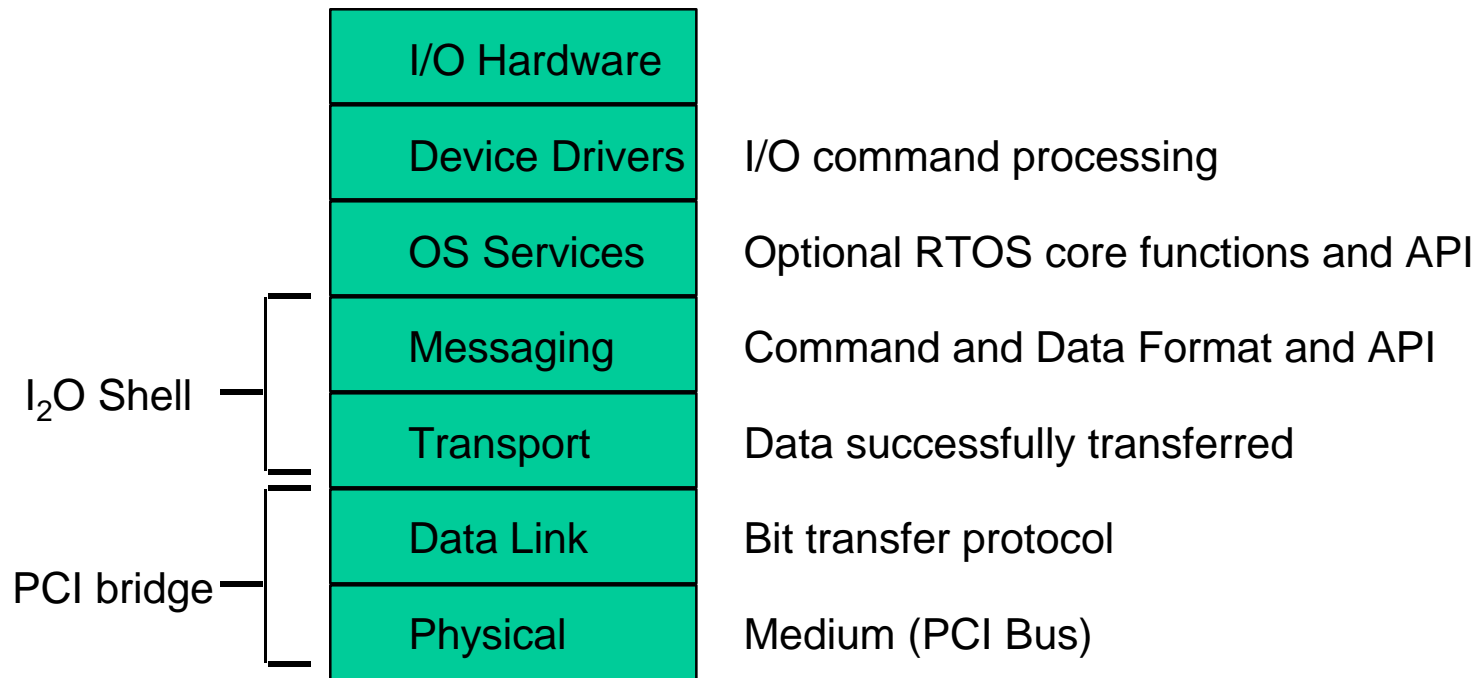
Benefits of I₂O

- ◆ Development of the I/O adapter hardware and software can be decoupled
- ◆ Improved testability and reduced test complexity
- ◆ Lower maintenance of device drivers
- ◆ Improved fault isolation and recovery
- ◆ Extensible to new CPUs, I/O technologies and interconnects
- ◆ Availability of off-the-shelf chip, software, and board components for I₂O systems

I₂O Software Architecture

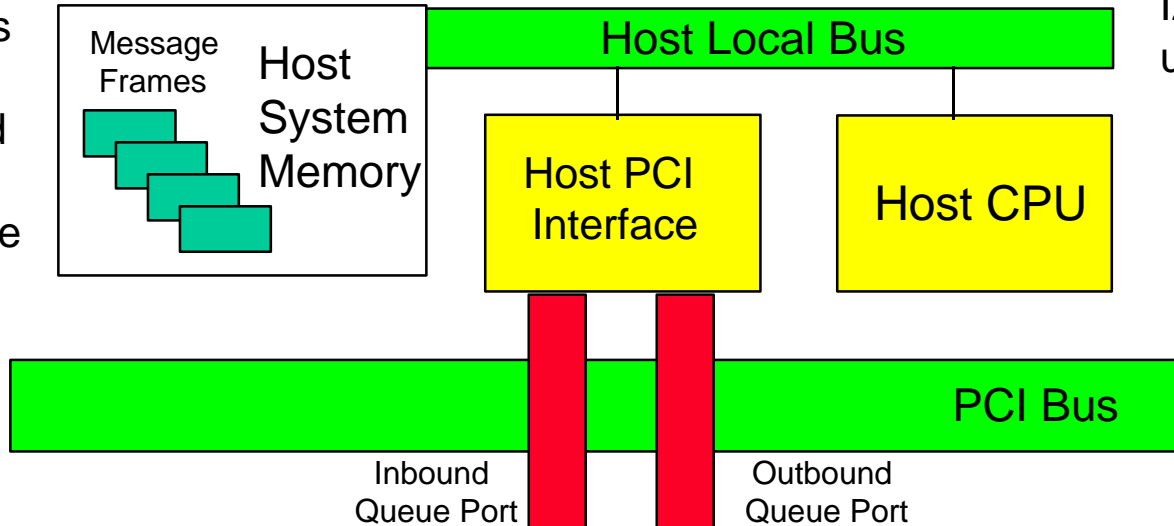
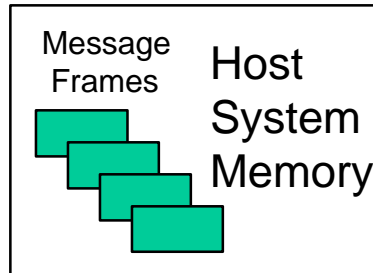


I₂O on PCI is a Typical Messaging Protocol Stack



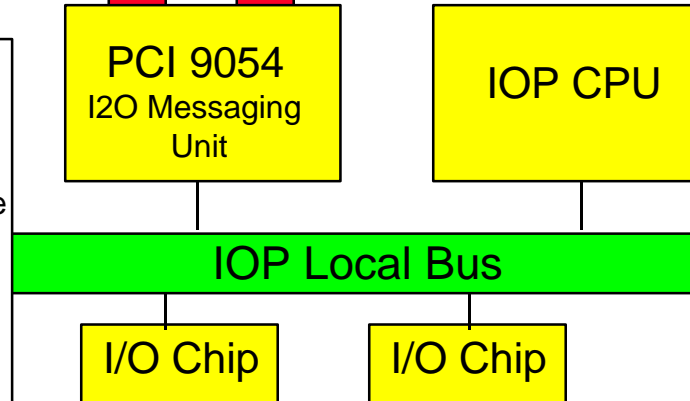
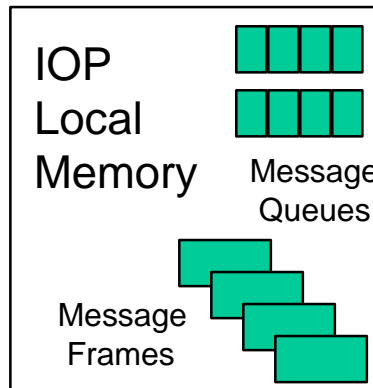
Building on PCI - Typical I₂O System Architecture

No hardware changes are required on the host side



I2O allows efficient use of:

- Bus Mastering
- DMA
- Interrupts
- Burst Modes
- Push/Pull memory



IOP Must Have:

- CPU
- Memory
- Messaging Unit



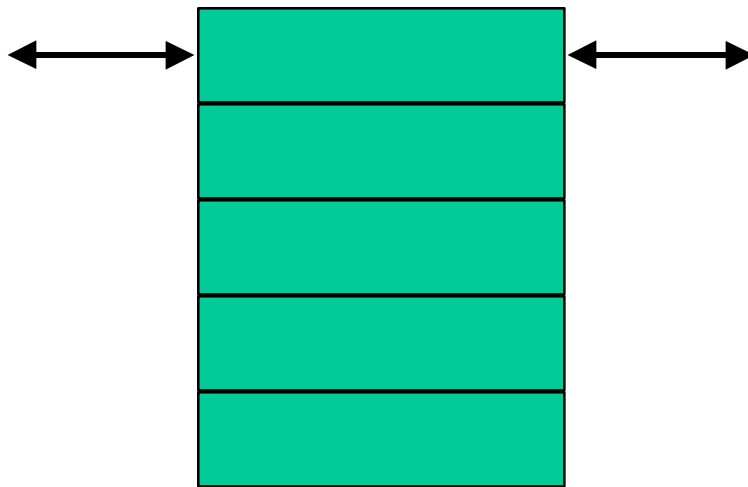
Message Passing Options

Choices for messaging protocol:

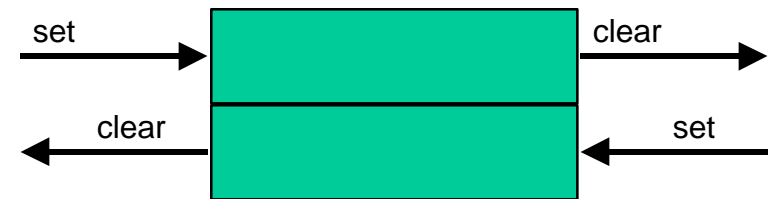
- ① Make up your own
 - ◆ Define commands, parameters, status
 - ◆ Define mailboxes, doorbells, interrupts
 - ◆ Define messages, buffers, queues
 - ◆ Define pointers, chaining, transfers
 - ◆ Define errors, recovery
 - ◆ Test the whole thing

Typical Message Passing Hardware Support

Mailbox registers can be read or written from both sides



Doorbell Registers Cause and clear Interrupts



Used for Passing:
Commands
Pointers
Status

Up to now, you got this, and did the rest yourself!

② Or Use I₂O (Intelligent I/O)



I₂O Defines it for You

- ◆ Defines commands, parameters, status
- ◆ Defines mailboxes, doorbells, interrupts
- ◆ Defines messages, buffers, queues
- ◆ Defines pointers, chaining, transfers
- ◆ Defines errors, recovery, testing

...and much more!

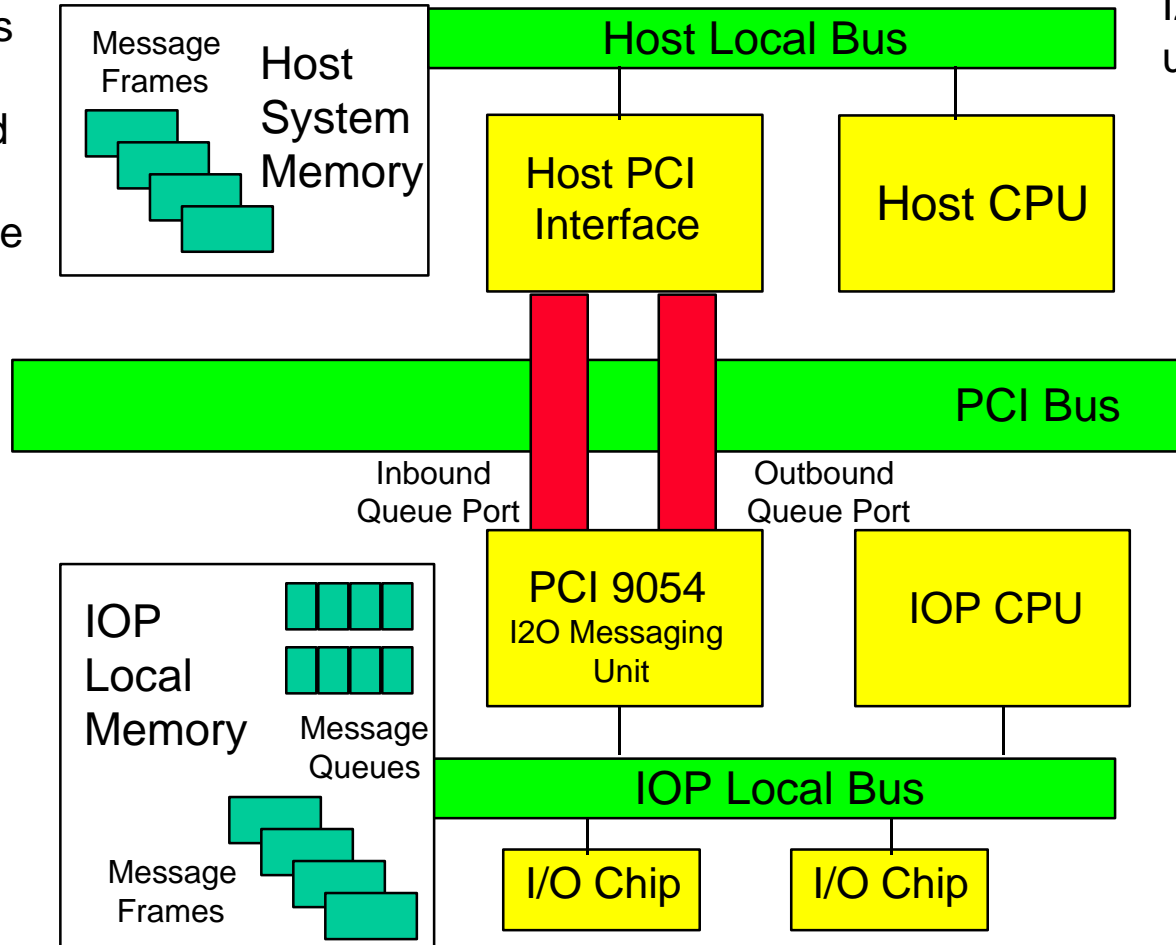
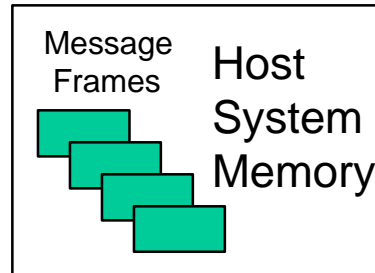


I₂O Messages and Classes

- ◆ Utility, Base, and Private Messages
- ◆ Executive Messages
- ◆ DDM Messages
- ◆ Initialization and Configuration Service
- ◆ Defined Command Sets (Classes)
 - ◆ Random Block Storage Class Tape Storage Class
 - ◆ SCSI Peripheral Class Fibre Channel Class
 - ◆ LAN Class WAN Class
 - ◆ IDE Class Floppy Class
- ◆ User Defined Messages and Classes

Building on PCI - I₂O Messaging Unit

No hardware changes are required on the host side



I2O allows efficient use of:

- Bus Mastering
- DMA
- Interrupts
- Burst Modes
- Push/Pull memory

IOP Must Have:

- CPU
- Memory
- Messaging Unit

When to Consider I₂O for a Private Platform Application

- ◆ Host CPU (motherboard or plug in card)
- ◆ One or more intelligent I/O subsystems with high speed I/O transaction devices
- ◆ Need efficient utilization of PCI bus
- ◆ Need improved compatibility and ease of upgrade to future designs
- ◆ Also: I₂O and non-I₂O devices can be used on the same PCI bus



Private Platform Applications

◆ Embedded Systems

- ❖ Network Routers/Switches/Remote access
- ❖ Telecomm Switches
- ❖ Printer Engines
- ❖ Industrial

◆ Adapter Boards

- ❖ Communications: Modems, ISDN, ATM, WAN, 10/100/Gigabit Ethernet, FDDI, Token Ring
- ❖ Disk Control: RAID, SCSI, Fibre Channel, SSA
- ❖ Video/Multimedia: MPEG I&II, DSP, Image Processing, DVD

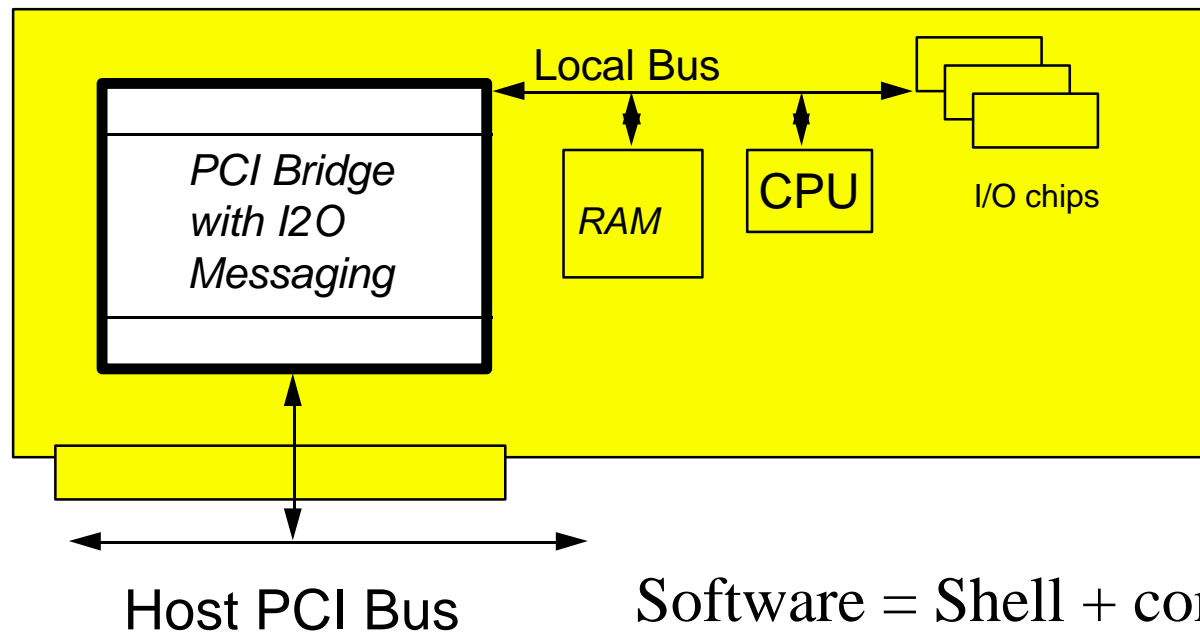


Private Platform I₂O Options

- ① Use I₂O PCI Bridge (e.g. PLX PCI 9054) with a RISC processor and shell software
 - ❖ Has been implemented with PowerPC, PowerQUICC, SuperH, MIPs, i960, ARM
- ② Use a dedicated IOP (e.g. PLX IOP 480) with shell software
 - ❖ Best price/performance for dedicated function

Private Platform IOP Design

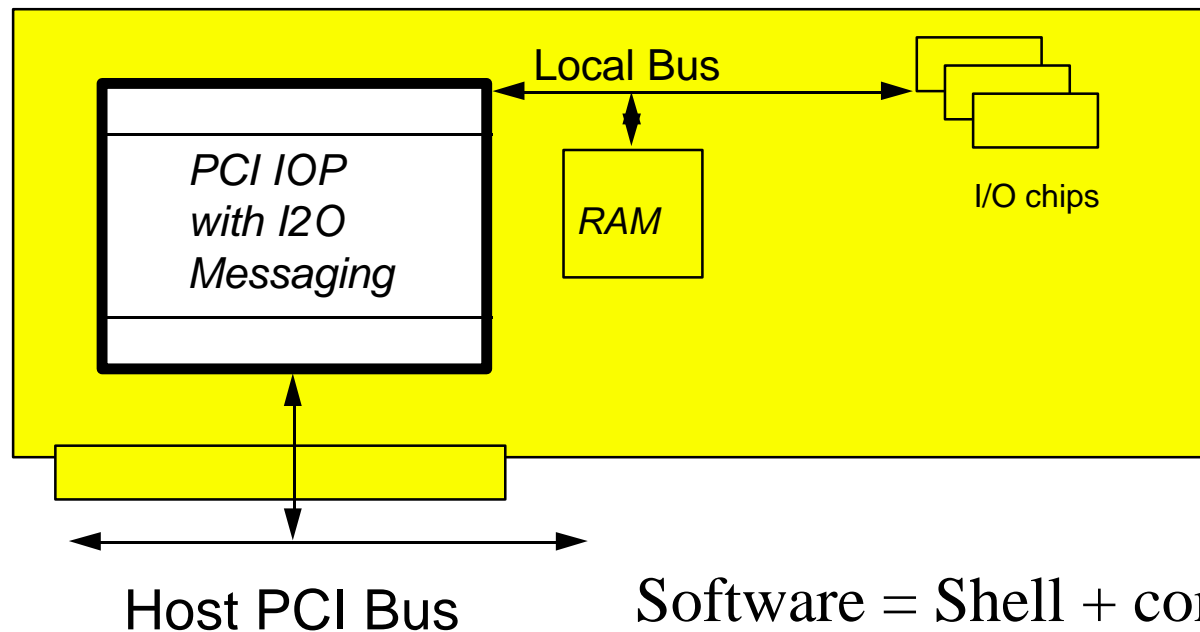
Form Factor can be PCI, Compact PCI, PMC or Other PCI



Software = Shell + command set
+control firmware (PDM)

Private Platform IOP Design

Form Factor can be PCI, Compact PCI, PMC or Other PCI



Software = Shell + command set
+control firmware (PDM)



PLX I2O Manager Shell ***for private platform design***

- ◆ I₂O v1.5 messaging software enables implementation of I₂O Private Platform designs on embedded systems using PLX PCI 9054, 9080, or IOP 480 chip
- ◆ Modular, and RTOS and CPU independent
- ◆ Allows minimum IOP implementation including I₂O shell, messaging protocol, configuration, and general purpose commands
- ◆ Includes sample RAMDISK Private Device Module (PDM) compatible with I₂O Mass Storage OSMs
- ◆ IOP developers create a custom PDM and link to I₂O Manager to create IOP
- ◆ Object code libraries for Intel i960J/C/H, IBM PowerPC 40x families, Motorola MPC 860 families



PLX I₂O Manager Overview

- ◆ Basic - Chap. 3
 - ◆ All basic including Utility, Base, Private Messages
 - ◆ Downloading of DDMs supported, but not “I₂O Loadable”
- ◆ I₂O Shell - Chap. 4
 - ◆ All
- ◆ Classes - Chap. 6
 - ◆ Executive Messages
 - ◆ DDM Message Subset
- ◆ Core - Chap. 5 Subset:
 - ◆ IOP State Table
 - ◆ IOP Initialization
 - ◆ DDM Initialization
 - ◆ Configuration Service
 - ◆ Message Service
 - ◆ Transport Service



PCI Bridges and IOPs

- ◆ PCI to Local bus bridges can include I2O messaging units
 - ◆ PLX PCI 9054 for PowerPC, SuperH, MIPs, i960
 - ◆ Intel 21285 for StrongARM
- ◆ PCI to PCI Bridge can have I2O messaging
 - ◆ Intel Drawbridge
- ◆ CPUs can have PCI and I2O messaging built in
 - ◆ PLX IOP 480 PowerPC
 - ◆ Intel i960Rx



IOP 480

Pumping Up the Pipe

CompactPCI

Hot Swap Friendly



V2.2, Power Mgmt & Vital Product Data

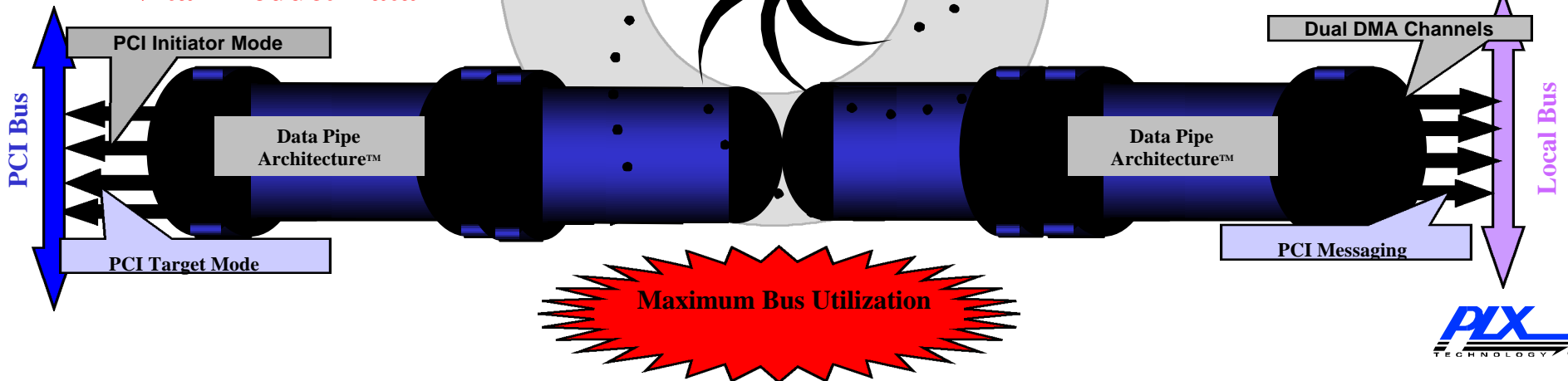
PPC Core 40x/60x
Code Compatible
66 MHz = 80 MIPS

3.3V Core/5v Tolerant
600mW Typical @ 66 MHz

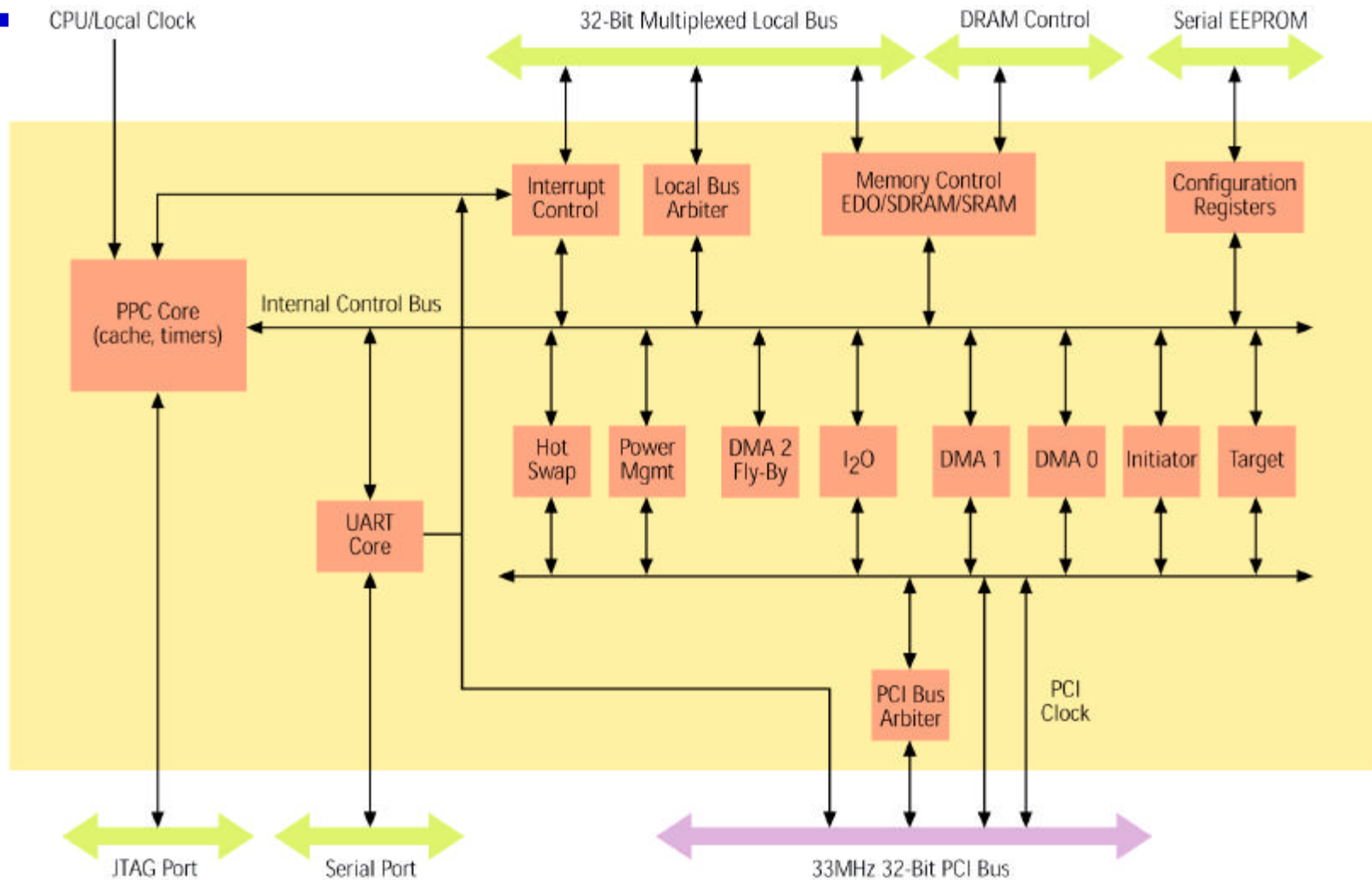
Space Saving
225 PBGA or 208 PQFP

Local Bus Fly-By DMA
Single Cycle I/O - Memory

Memory Controller
Supports 256MB
EDO/SDRAM/SRAM



IOP 480 Block Diagram





Summary

- ◆ I₂O brings many benefits to designs that need to use intelligent I/O
- ◆ Using I₂O as a message passing protocol will save development effort and time
- ◆ A staged approach can be taken to minimize disruption and risk
- ◆ I/O processors further simplify the implementation of distributed processing architectures and I₂O



www.plxtech.com

