On Chip Standard Hot-Plug Controller

January 15, 2007 Version 1.0

1 Introduction

Today, businesses and consumers are equally dependent on the Internet infrastructure and around-the-clock data availability. The infrastructure designed to support 24/7 services includes redundant systems (or subsystems) to avoid single point of failure and provide the ability to replace failing subsystems without power down or disruption to the service. Additionally, many systems require the capability to add new subsystems (or modules) to meet increasing demand without service interruption (power down). The PCI Express (PCIe) specification adopted the PCI standard Hot-Plug controller specification (SHPC) to support this need in PCIe based systems. The SHPC specification defines various status registers for host CPU access and signals/pins to help gracefully power up/down the subsystems being replaced or added.

2 On-Chip SHPC Support

Chip vendors developing PCIe switches and bridges have taken two approaches: on-chip or off-chip SHPC support. In the on-chip SHPC approach, the burden of controller logic and additional signals is carried by the chip provider. In the off-chip SHPC approach, the chip vendor pushes the problem of implementation down to system hardware designers. The off-chip approach requires additional circuitry (PLD/FPGA) on the board, hence, increasing the cost of the material, board space required and complexity of the design. This also impacts the CPU performance as it has to communicate with slow PLD/FPGAs. Figure 1 illustrates this approach.

PLX Technology has taken the on-chip integrating approach for SHPC support on all of its PCIe switches. In a PLX implementation, all required Hot-Plug status registers are integrated on the chip and all required signals for Hot-Plug implementation are provided. PLX switches support from three to eight ports with Hot-Plug capability. This substantially reduces the cost and complexity of implementation for systems needing SHPC functionality. Figure 2 illustrates the PLX approach.

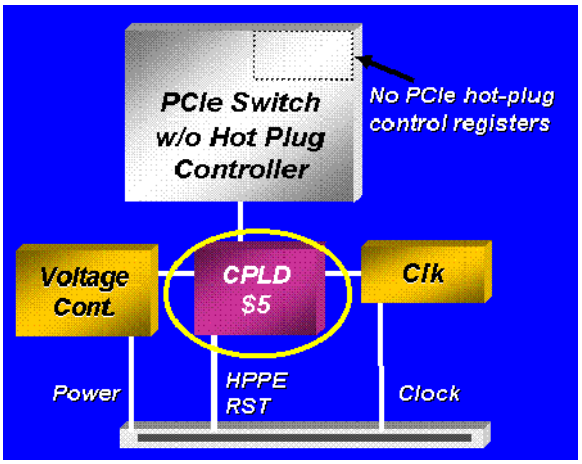

Figure 1

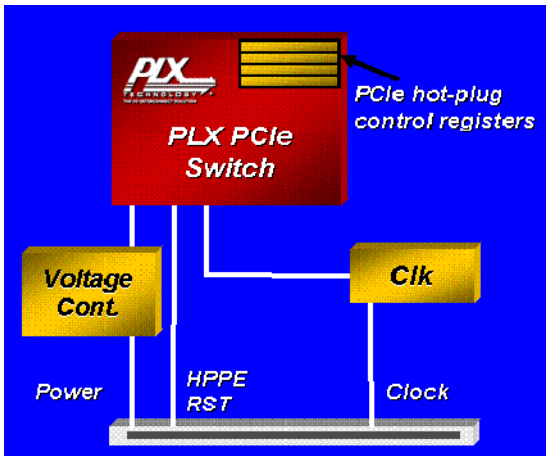

Figure 2