DISTRIBUTED PLD SOLUTION FOR REDUCED SERVER COST AND INCREASED FLEXIBILITY

A Lattice Semiconductor White Paper

August 2017

Lattice Semiconductor

111 5th Ave., Suite 700

Portland, Oregon 97204 USA

Telephone: (503) 268-8000

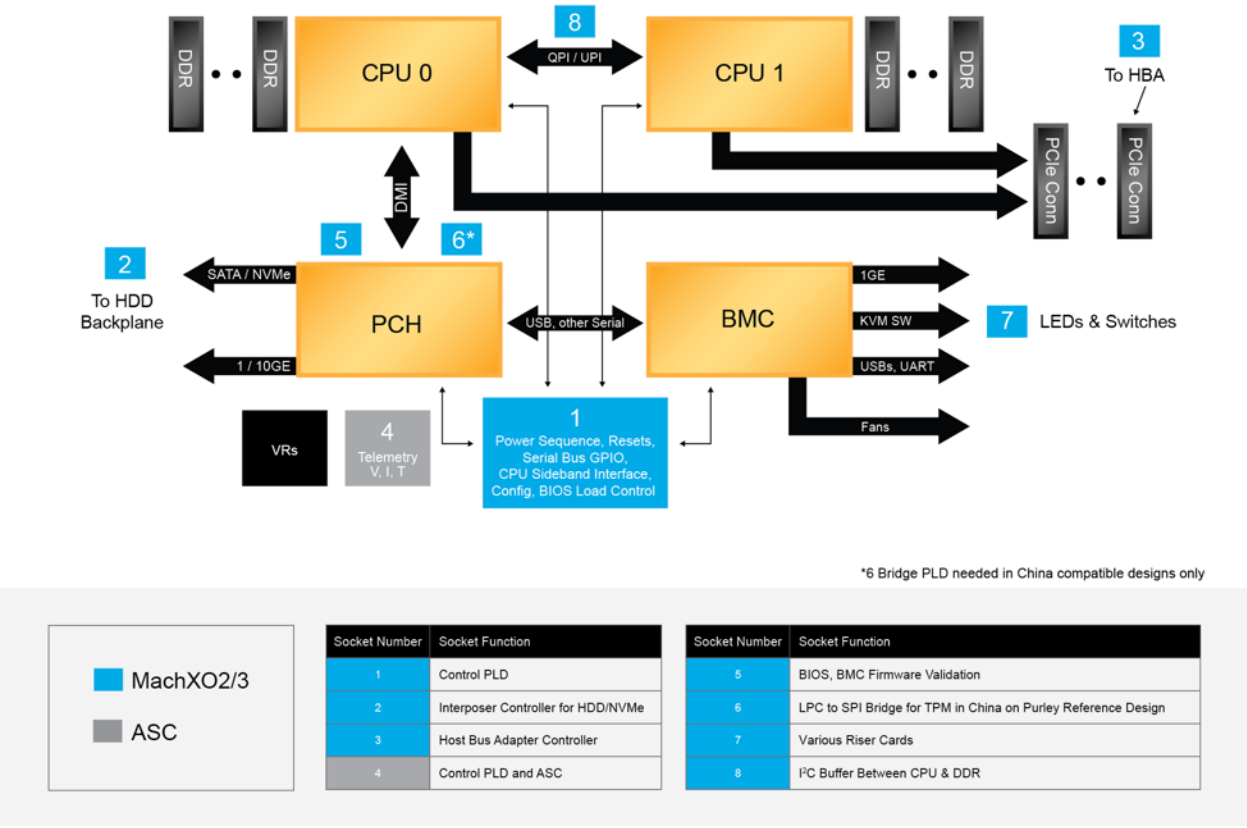
www.latticesemi.com

Introduction

Servers come in many different types: From rack and blade versions to tower and modular configurations for high density computing. Ideally each server is optimized to perform its specific task. On closer observation, however, most server designs share a number of common characteristics. Typically, they feature multiple processors, hot swappable storage, wide range of peripherals connected to CPU and PCH via PCIe, security services, and power management resources to name just a few common elements. So, while designers appear to create very different solutions for various applications, in most cases, they are customizing a basic server architecture.

Fig. 1 below illustrates this common architecture. More often than not, server designers customize this basic architecture to meet the needs of different markets. The use of peripheral hardware blocks, system level interface blocks, BMC interfaces, and other key components may vary from one server design to another. However, the power management, control and glue logic function (shown as Socket Function #1) block consistently plays a key role in the customization of a design to meet specific application requirements. Designers need to modify functions such as power management, board specific glue logic, or I/O expansion for each server type. Although Socket Function #1 does not play a role in any of the payload functions such as CPU, hard disk or networking, it is needed to make all the major devices on the board function within their operational limits. Consequently, designers are constantly trying to reduce the cost and complexity of these functions without trading off the board reliability.

This paper discusses the traditional approach of implementing Function #1 in older generation server designs and compares them with the approach used in modern server designs, where PLDs are used to integrate these functionalities. In addition, this paper discusses other functions commonly found in servers integrated into other programmable devices in order to reduce complexity and cost.



Socket Number	Socket Function
1	Control PLD
2	Interposer Controller for HDD/NVMe
3	Host Bus Adapter Controller
4	Control PLD and ASC

Socket Number	Socket Function
5	BIOS, BMC Firmware Validation
6	LPC to SPI Bridge for TPM in China on Purley Reference Design
7	Various Riser Cards
8	I²C Buffer Between CPU & DDR

Fig. 1: Server block diagram with 8 PLD Socket Functions (use cases)

Socket Function #1 – Power Management, Control and Glue Logic Functions

Historically, designers have typically opted for implementing the power management, control and other glue logic functions using many types of discrete components. For many years, that approach offered the more cost effective path. But as server designs have grown increasingly complex and the number of functions rose proportionately, discrete designs started to require larger and larger numbers of devices. Today, designers who are using the discrete approach end up spending more time and resources to design the Socket Function #1 block for multiple server types. For example, changing of the number of complex SoC devices on the board may result in altering the number of supplies, glue logic and other control functions. This may warrant significant changes to the logic and the underlying timing.

Consequently, the use of discrete device solutions not only delays the release of newer types of server hardware, but also increases the cost as the number of components

required for implementation grows. In addition, design changes sometimes require a re-spin of the entire circuit board, which further delays the project and adds cost.

Modern server systems typically integrate Socket Function #1 into non-volatile PLDs. These PLDs are expected to commence their operations as soon as the power to the board is applied (instant-on). Typically, the logic density and the number of I/Os required to implement Socket Function #1 depends on the server type. Consequently, a PLD family rich in I/Os and density options is ideally suited for implementing Function #1.

Lattice's MachXO3 FPGA family, and its predecessor, the MachXO2 family (referred to as MachXO2/3), both deliver those capabilities. The MachXO2/3 devices are instant-on, non-volatile PLDs, ranging from 640 LUTs up to 9400 LUTs and offer from 22 I/Os up to 384 I/Os. These PLDs can be transparently updated in the system and offer Dual Boot to recover from any in-system update errors. These devices only need a single 3.3 V supply to operate and the server board power management algorithm starts to become operational when the 3.3 V supply is above 2.2 V. As a result, the MachXO2/XO3 is the first device on the board to turn on and the last device to turn off. These devices support multiple I/O banks that can be powered on or off individually without affecting the operation of other blocks. This enables them to integrate multiple heterogeneous functions, such as multi-power domain control, out-of-band signaling, and power stand-by control. They also offer designers the ability to add SPI, I^2C and timer/counter interfaces to legacy designs, and support multi-time programmable on-chip configuration Flash memory. Finally, these state-of-the-art devices are available in 5 mm x 5 mm QFN and BGA packages with 1 mm and 0.80 mm ball pitch.

Function #1 Integrated into a Control PLD (MachXO2/3)

In Fig. 2, the MachXO2/3 device is used for the implementation of control PLD functions, such as power/reset sequencing, various types of serial busses (I^2C, SPI, eSPI, SGPIO, etc.), debug ports, LED drives, FAN PWM driver, front panel switches sensing and other general GPIO functions. The MachXO2/3 devices support 1V signaling, which enables them to perform out of band signal integration without the need for external GTL transceivers. Lattice's software package tool, Reveal, can be used to debug the control PLD circuit, while the chip is functioning. Running on a PC, this tool can be considered a logic analyzer for monitoring and capturing of various states, leading up to a fault event. For example, the Reveal debug tool enables designers to capture a number of event traces (comprised of registers, nodes and pins states)

leading to the faulty condition and displays them on a PC monitor. This significantly reduces the board debug time of their system.

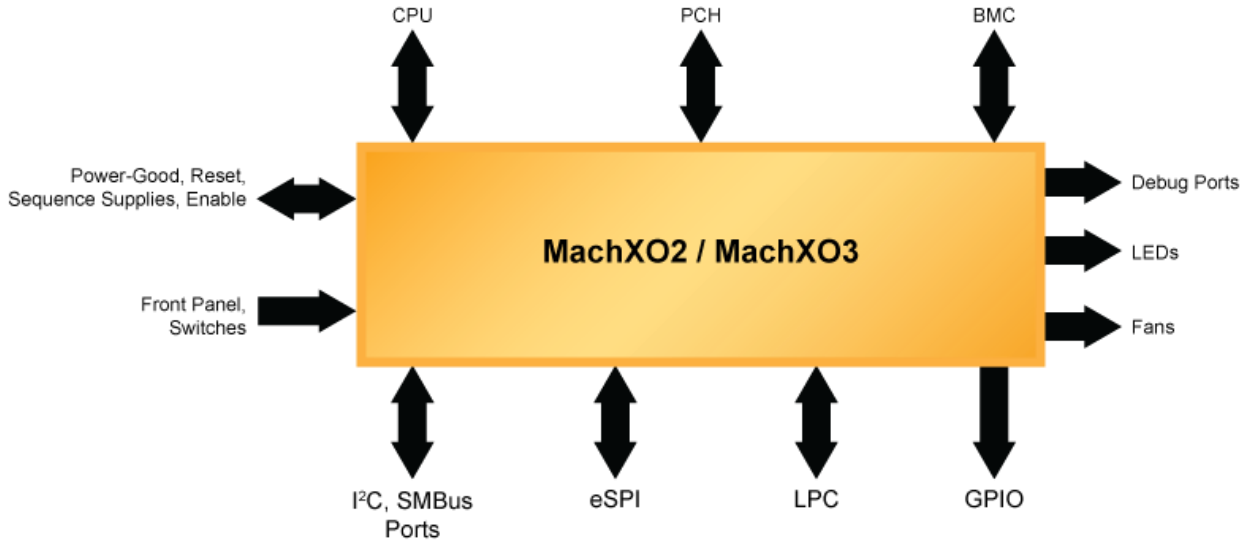


Fig. 2: Control PLDs based on the MachX02/MachX03

Hitless I/O

Control PLDs enable the designers to significantly reduce time-to-market and enable them to meet the market pressures of bringing out a new customized hardware within the allotted time. Sometimes, there could be bugs in the implementation of the control function or the overall system architecture that may require a new function. A common approach to accomplish a modification to the design is through an in-system update and power cycling of the system to bring the newly programmed image into service. This act of power cycling interrupts the operation of the entire server hardware, reducing its availability. To ensure the continuous operation of high availability systems, the MachXO3 devices can hold the I/Os unchanged, while the configuration refresh occurs and the new configuration initializes. This feature is called Hitless I/O.

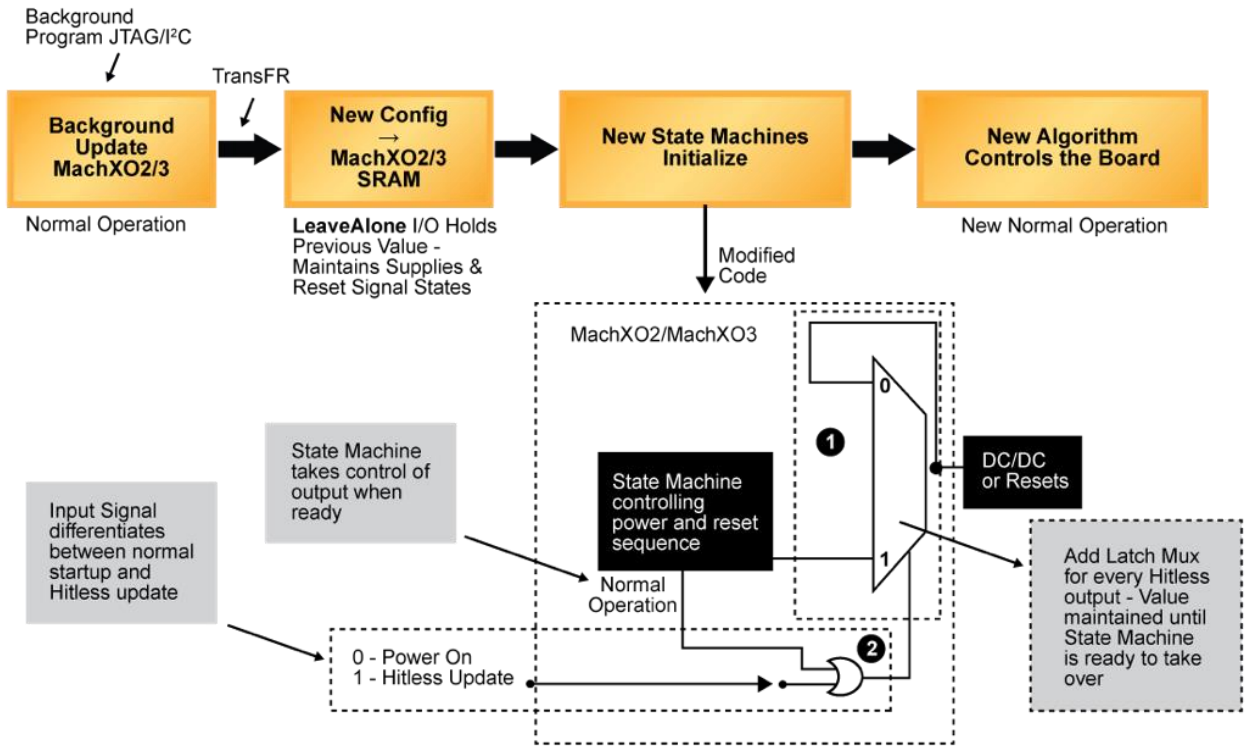


Fig. 3: How Hitless I/O works

Hitless I/O Operation (Fig. 3)

To enable zero-downtime updates, the MachX02/MachX03 devices undergo a "background update" that loads new configuration data into its configuration Flash memory. When the upload is complete, a "TransFR" command moves the new PLD image from the configuration Flash memory to the PLD's configuration SRAM. At the same time a "Leave Alone" function ensures that all I/O values are held in their last known value. Finally, during the "Logic Initialization" stage, the state machines begin to restart the power management and reset distribution functions, which results in turning the power supplies off and forces the board to undergo power recycling.

How does the system hold the outputs controlling the supplies and other logic control signals, while the state machines created by the new image undergo initialization? To keep the critical I/O unchanged during the initialization process, Lattice adds a latch MUX to every critical I/O. These elements hold the outputs at their last known value during the state machine initialization process and, once the process is complete, pass the output control back to the state machines. Key to this process is the circuit's ability to differentiate between a normal (power-on) startup and after a reconfiguration event using a separate "Hitless_I/O_Enable" input that prevents a lock of critical outputs during a normal power-on sequence.

The advantages of this new capability are tremendous. It gives manufacturers the flexibility to implement on-the-fly configuration changes to correct design flaws or add new capabilities to their products. It can also be very useful during product development as it allows designers to quickly turnaround a product during debugging or create specialized product variants during a rack installation procedure.

The convenience and cost advantages of PLDs make them ideal for supporting in-system design updates and manage power supplies, monitor and control critical signals, while performing basic housekeeping functions.

Socket Function #2 – Logic Functions Needed to Support Hot Swappable Disks

Rack servers support hot swappable HDD/FD/NVMe drives. These disk drives are plugged into a back plane. The back plane interfaces to the main mother board through serial interfaces, such as SGPIO and I^2C. A MachX02/3 device can be used to integrate the logic function like the one depicted in Fig. 4 to provide backplane control. For example, when an NVMe drive is plugged into the drive slot, the logic in the MachXO2/3 device will automatically route the status and control signals to I^2C bus instead of SGPIO bus.

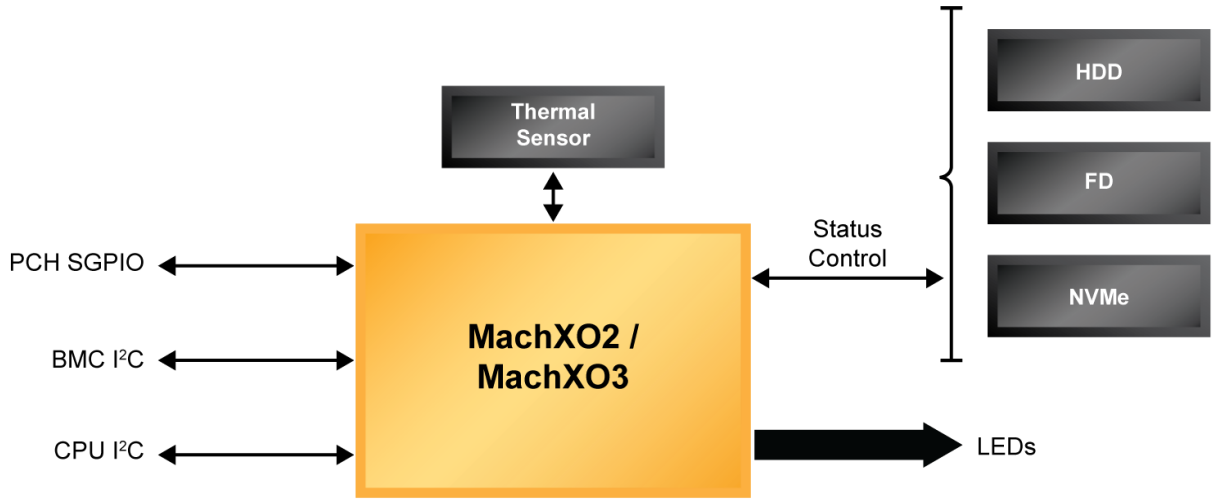


Fig. 4: Simplifying backplane control of hot swappable drives using MachXO2/MachXO3 PLDs

Socket Function #3 – Hardware Management of Host Bus Adapter Board

Another potential application for Lattice's MachXO2/3 devices lies in the integration of host bus adapter control logic. As indicated in Fig. 5 below, this solution integrates SGPIO and other out-of-band signaling, manages power/reset sequencing and other PLD functions, including fast supply fault detect, and status save. Designers can also add features and bug fixes to the logic implemented in the MachXO2/3 device in the field, without interrupting system operation via the Hitless I/O feature and an I^2C interface.

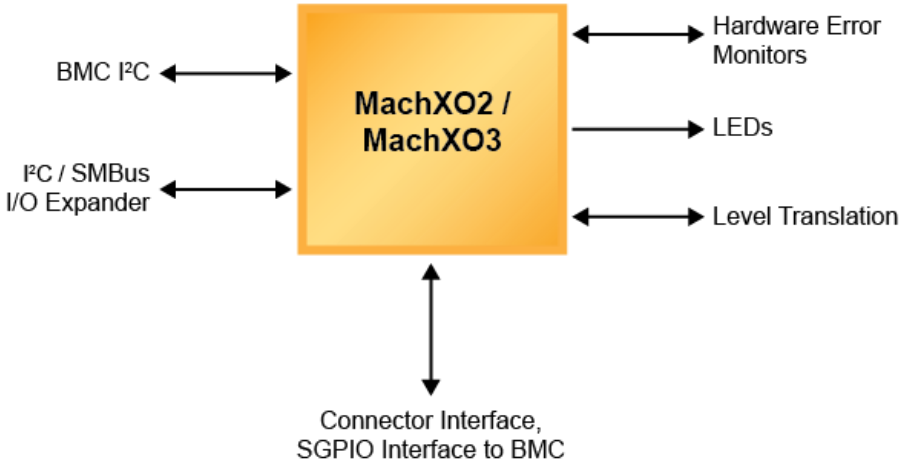


Fig. 5: Integration of host bus adapter logic

Socket Function #4 – Voltage Current and Temperature Telemetry

Typically, systems measure onboard supply voltages, board and device temperatures and current load on some important supplies on an ongoing basis. To assist in the measurement of these parameters server boards use Analog to digital converter ICs to supplement the number of channels integrated into the BMC, external temperature sense ICs and current sense ICs. In addition, the board uses I^2C buffer ICs and I^2C multiplexer ICs to manage the telemetry I^2C busses (Fig. 6). The DC-DC converters on the board are used to supply power to the ICs, are controlled by the Control PLD device. The Control PLD also monitors the 'Power-Good' digital signals from the DC-DC converters.

Designers can take advantage of the Lattice's ASC (Analog Sense and Control) device in conjunction with the control PLD to integrate the ADC IC and some of the temperature sense ICs. At the same time, the device transfers the "Enable" and "Power-Good" signals from the control PLD to the ASC devices. This frees up I/Os on the

control PLD, which then can be used to integrate I²C buffers and I²C multiplexer ICs functions. This results in overall reduced cost and BOM of the telemetry circuit. Additionally, by sensing the supply voltages for “Power-Good” condition, as well as power-off condition, ASC also helps improve the reliability of power down sequencing and minimizes circuit board congestion.

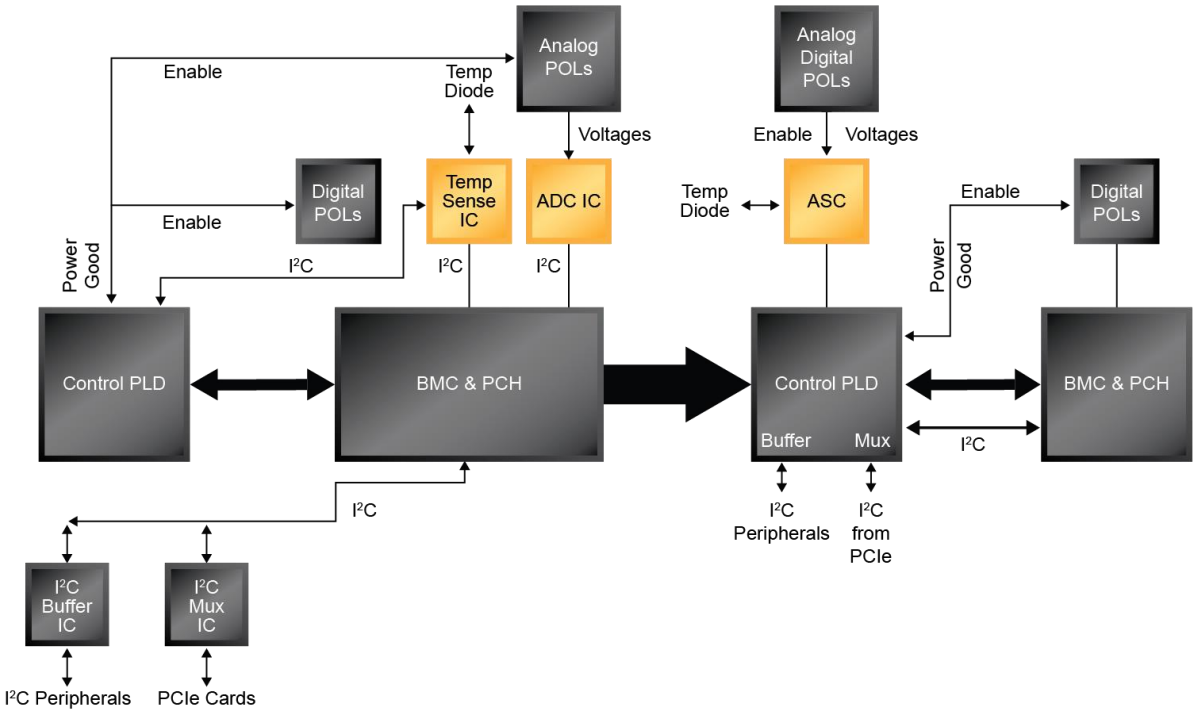


Fig. 6: New approach to integration of telemetry functions

Socket Function #5 – Bios and BMC Firmware Authentication

To help ensure BIOS and BMC firmware authentication, a MachXO2/3 device can serve as a hardware root-of-trust security (Fig. 7). In this configuration, these devices can be used to validate the system BIOS and BMC firmware using Elliptic Curve Signature Authentication. They can also be used to manage automatic golden image switchover in the case of a compromised active image.

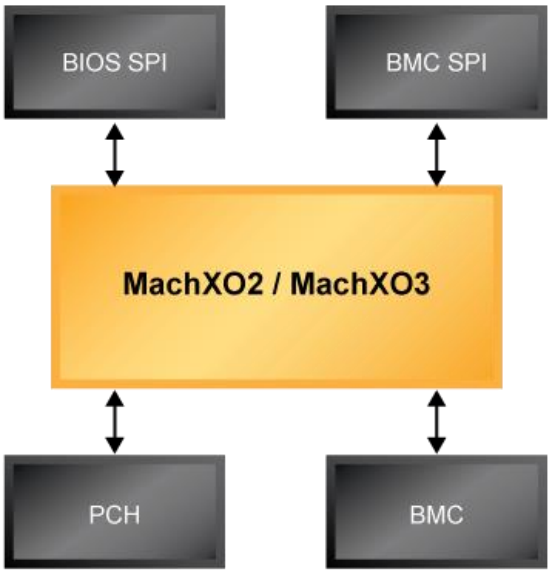


Fig. 7: MachX02/3-based solutions manages and validates BIOS and BMC firmware authentication

Socket Function #6 – Bridging Between TPM/ TCM and Single SPI Interface on PCH

Lattice's MachX02/3 devices offer extensive bridging capabilities. For example, server designers can use these devices to bridge between a PCH SPI interface with a TPM module (used in countries outside China) or with a TCM module (used in China) on the same hardware (Fig. 8). This bridge is compatible with a wide range of operating frequencies at ingress and egress points.

Fig. 8: LPC to SPI Bridge for TCM.

Socket Function #7 – Integrating Multiple Functions on Riser Cards

Often servers use riser cards to connect LED drive, control, and enclosure sense function on a riser card to reduce the number of interconnections on the main board. Often, these functions are implemented using discrete logic ICs, which results in multiple types of riser cards, each with slightly different functionality. An option to reduce the number of riser card types is to integrate the functions for each of the cards onto a MachXO2/3 PLD. One can then customize the logic on the card by simply modifying the logic integrated in the MachXO2/3 device during manufacturing.

Socket Function #8 – Integrating Multiple I^2C Buffers

The CPU in a server system communicates with the DDR memory DIMMs via a pair of I^2C buffers (Fig. 9). The CPU also monitors the SSD drive through another I^2C interface. Designers are required to use voltage level translator buffers to map CPU's 1.05 V I^2C interface with the DDR memories operating with 1.2 V and the SSD drives operating at 3.3 V. The CPU also generates multiple out-of-band signals using 1.05 V logic signal interface. These out-of-band logic signals are required to communicate with other devices operating with a signal interface of 2.5 V or 3.3 V. This requires the use of GTL buffers on the board.

Low cost MachXO3 devices in a small QFN package (5 mm x 5 mm) can be used to integrate level translation from 1.05 V I^2C and other logic signals to 1.2 V, 3.3 and 2.5 V. This reduces the circuit board area, BOM and more importantly cost of implementation of this functionality.

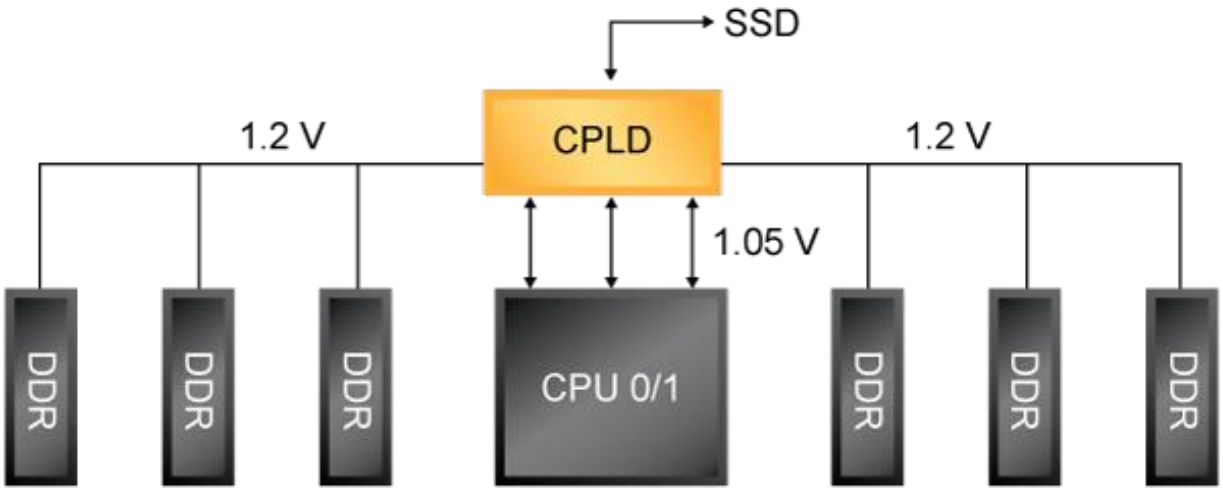


Fig. 9: MachXO2/O3 1V I/O supports the integration of I^2C buffers

Conclusion

Like their predecessors, today's server designers are constantly trying to pack more functionality on their boards as quickly and cost-effectively as possible. One often overlooked strategy to accomplish that task is the implementation of control PLDs. By offering designers a simple way to integrate all control path functions into a single programmable device, and by adding new capabilities that allow designers to modify designs even after they have shipped to the field, control PLDs promise to significantly simplify board design and debug.