



The right gear

Bit based dynamic alignment for multi gigabit parallel I/O. By **Shakeel Peera**.

Whilst I/O standards continue to evolve towards serialisation in backplane and, more recently, chip to chip applications, high speed parallel I/O still has an important role to play where serial technologies are cost prohibitive.

Apart from established source synchronous I/O standards – such as SPI4.2, SFI4.1, XGMII, HyperTransport, Rapid IO and CSIX – the next generation of clock forwarded interfaces is being implemented on sram and dram (DDR1/2/3, RDRAM1/2 and QDR2), and in a/d and d/a converters. The I/O speeds required for these next generation applications are expected to exceed 1Gbit/s.

FPGAs are increasingly used as programmable SoCs, designed as an integral part of the system data path. However, the expectation is these devices can perform high speed I/O translation and processing. And users expect them to comply with past, existing and emerging I/O standards.

Although electrical compliance and high speed signal integrity are required features, these alone do not address the bandwidth issue. The fpga I/O must also have circuitry to manage and maintain the clock and data relationships of these high speed signals, as well as providing the necessary 'gearbox' functionality to support the transfer of high speed I/O data to the fpga fabric to perform the required processing.

FPGA vendors have tackled this problem in various ways, using pre engineered, dedicated I/O circuitry to perform specialised functions. The objective is to provide a transparent scheme that maximises

performance yet minimises the need for board level 'tweaks'.

The LatticeSC Purespeed I/O architecture addresses some of these concerns by delivering a range of source synchronous I/Os.

Embedded I/O logic

High speed source synchronous interfaces pose challenges for the designer:

- Managing and maintaining the clock to data relationship
- Managing the data to data skew (word alignment)
- Clock domain transfer of these high speed signals to the fpga fabric

Word alignment and deskew is fairly straightforward and can be handled by fpga logic, but the clock to data relationship and clock domain transfers are more challenging.

For bit and bus deskew, designers have traditionally relied on methods such as matching bus trace lengths, or manipulating the clock signal with PLLs or DLLs, eliminating clock injection delay and/or phase shifting

the clock by a predetermined percentage to maximise the clock to data relationship.

These methods have shortcomings, since they are static and don't account for variations that can occur over process, voltage and temperature (PVT). Today's high speed interfaces require bit based compensation due to the increased difficulty of meeting and maintaining adequate setup and hold time margins for shrinking clock cycle times.

This issue is exacerbated for high speed parallel protocols, such as SPI4.2, in which dynamic bit based alignment and word alignment are key. Figure 1 shows the effects of dynamic bit alignment. Without compensation, the clock is not centred on the data eye and barely meets the set up time for bit A on channel 0.

Even worse, the incorrect data bit is sampled on channel 1. When bit based compensation is applied, the clock is centred and the

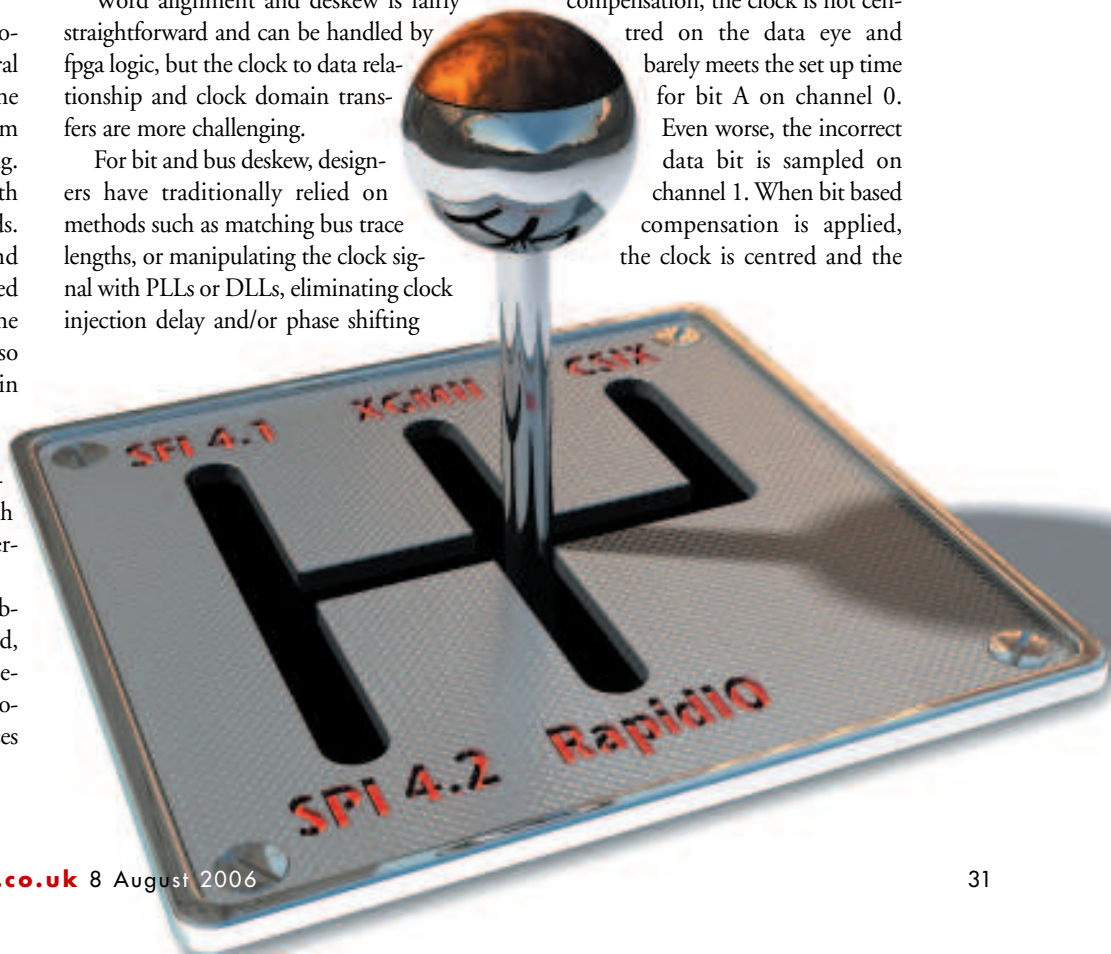
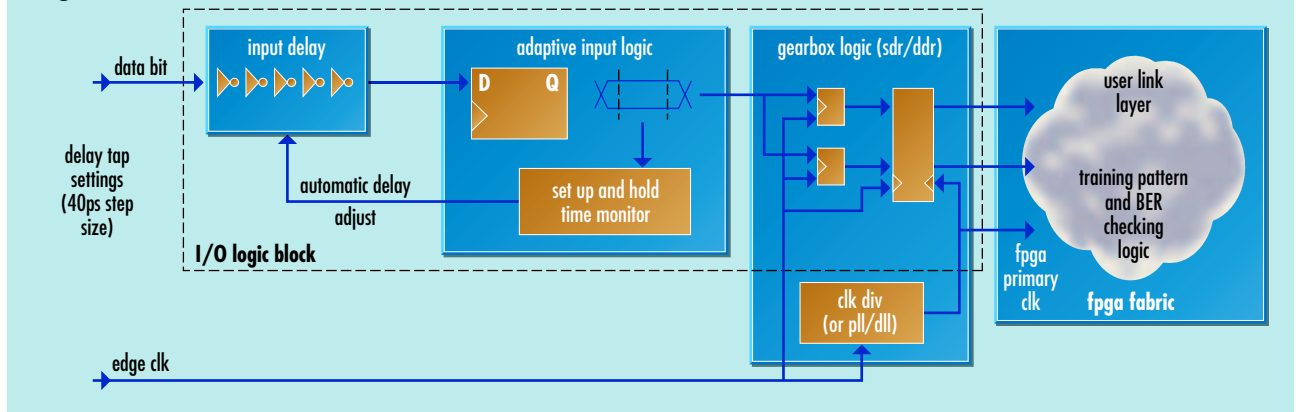




Figure 2: Purespeed I/O bit based dynamic alignment mode



proper bit sampled for both channels.

Once the data is sampled, the task of transferring the high speed clock and data to the fpga fabric still remains. In this case, 'gearbox' logic is required to ensure the signals can be passed seamlessly to the fpga at the lower speeds suitable for further processing.

Gearbox logic

The critical issue here is that all bits on the high speed bus must be transferred to the low speed bus on the same low speed clock edge, requiring the synchronisation to be timed using the high speed clock. This synchronisation must be built into the fpga as dedicated logic in the I/O block since it cannot be done reliably at high speeds using fpga gates. The Purespeed I/O logic block simplifies this process, providing a flexible and proven I/O system.

Whilst PLLs and DLLs can be used to align data and clock, a simple alternative when the clock to data relationship is known is to use an input delay block.

However, for higher speed interfaces,

bus based alignment does not provide the precision necessary because the delay compensation is applied globally. What is needed is a closed loop control and monitoring circuit that maintains proper setup dynamically and holds time margins on a bit by bit basis. For this reason, the Purespeed I/O block has a mode in which the input delay block is used in conjunction with embedded adaptive input logic to tackle high speed applications in which the clock to data relationship is unknown.

Bit based dynamic alignment (INDEL+ AIL) is the most robust configuration (see Figure 2). Users can establish and maintain the clock to data relationship dynamically on a bit by bit basis, providing the resolution necessary to support speeds of up to 2Gbit/s on a single pin.

The key to this mode is that it is a self contained, closed loop system that can be enabled/disabled or updated under fpga control. The closed loop design also allows for tracking and compensating for delay variations due to process, voltage and temperature conditions.

The high speed nature of these interfaces means gearbox logic must be used to slow these signals to speeds manageable by the fpga fabric. As shown in Figure 2, the Purespeed I/O block provides this gearbox logic for either SDR or DDR interfaces.

On die clock dividers support the clocking requirements of the gearbox logic, alleviating the need for generic PLL/DLL resources. Another feature of gearbox logic is to provide the proper domain transfer of the high speed edge clock to the lower speed fpga system clock, guaranteed across process, voltage and temperature. Although an input example is shown, gearing logic is available for outputs.

Parallel I/O continues to have an important role in contemporary systems and demands are placed on FPGA architectures to deliver multi gigabit I/O performance. Demand can be met when an fpga has an I/O architecture designed as a true system level device and takes into account features necessary to deliver a true system level solution. These features must include best in class I/O buffers, termination technology that delivers linearity and signal integrity, I/O logic that deals seamlessly with dynamic clock and data alignment, and clocking and gearing resources that manage the processing and transfer of high speed signals to the fpga fabric without requiring fpga resources.

Author profile:

Shakeel Peera is strategic marketing manager with Lattice Semiconductor.

Figure 1: Parallel bus skew and the effects of bit based dynamic alignment

