White Paper

The Evolution of High-Speed Transceiver Technology

Introduction

The Internet revolution has led to a massive increase in data traffic. This trend is set to continue; over the next few years and it is likely that 95% of all communication traffic will shift to data. The need to support high bandwidth traffic has required that equipment performance grow at an exponential rate. WAN equipment which only two years ago operated at speeds of OC-48 (2.5Gbps) now runs at speeds up to OC-192 (10Gbps), and the development of OC-768 equipment (40Gbps) utilizing System Packet Interface-5 (SPI-5) and SERDES Framer Interface-5 (SFI-5) is already underway. System performance has also increased to support this infrastructure. Backplane and chip-to-chip interfaces supporting multiple serial lines of data are replacing parallel bus implementations in many applications. For example the 10 Gigabit Ethernet XAUI protocol is becoming increasingly popular in backplane applications, where four aligned channels of 3.125 Gbps collectively provide a backplane data rate of 10 Gbps.

Increases in performance have led semiconductor vendors to develop products capable of handling data rates in excess of 40 Gbps. This in turn has caused a continued evolution from single-ended I/O to differential signaling and the use of serialization and embedded clock recovery.

The latest deployed systems use transceivers capable of supporting I/O speeds up to 3.125 Gbps. At these speeds the transceiver block must include more efficient clocking circuitry employing encoding/decoding, clock rate matching, and alignment techniques to ensure accurate transmission of data through multiple media. This white paper discusses the evolution of this transceiver technology and describes areas of application where it may be used.

Transceiver Technology

Since the advent of the computer there has always been a requirement to send data. Data inside the system has traditionally been passed in a parallel form. Once data is transmitted out of the system, cable sizes required to carry the data and crosstalk issues associated with multiple signals in close proximity make parallel data transfer problematic. Serial transmission was adopted as a solution that simplified data transmission protocols, cabling schemes and system layout. The first schemes were based around single ended transmissions.

As transceiver technology expanded, industry standards emerged. One of the first transmission standards originating in the 1950s was RS232. This widely used single ended interface operates from a ±12 V voltage supply to provide a +24V signal swing to overcome noise and attenuation issues. The EIS-RS232 V.24 standard limits transmission characteristics to 15 meters distance at 20 Kbps, but deployments of this standard often operate over greater distances and lower baud rates.

Transistor to Transistor Logic

Transistor to Transistor Logic (TTL) has become the most common and basic of I/O standards. TTL operates from a +5V power supply and can transmit at speeds up to 100 MHz. However TTL tends to require high quiescent current particularly in the faster device families. Initially integrated circuits also employed TTL as the basic building blocks for design and a number of Application Specific Standard Products (ASSP) such as microprocessors and memory devices were built with TTL logic. During the 1980's Complementary Metal Oxide Semiconductor (CMOS) devices became popular, particularly for large scale integration because of their low (zero) quiescent current, good noise immunity and lower cost of manufacture.

TTL performance can suffer in certain implementations because device transistors are required to be either saturated or completely off. This requirement causes switching to be slow because the load capacitances take time to either fully charge or discharge. The introduction of Low Voltage TTL (LVTTL) greatly improved performance (and

reduced power consumption) by reducing the voltage swing required to change state. LVTTL supports higher data rates and has been widely adopted in high speed board design.

Emitter Coupled Logic

Emitter Coupled Logic (ECL) was used as an alternative to traditional TTL logic because it is better suited for high speed data transmission. ECL transistors always operate in the active region and therefore do not suffer from the same transistor switching issues found in TTL. A small DC bias is placed on the base of the transistor, resulting in a small voltage swing from base to emitter, and allowing the device to change state more quickly. The device relies on the emitter current flow for operation rather than the precise voltage levels used in TTL, causing a greater current draw. As a result ECL is also known as Current Mode Logic (CML) due to the high amounts of current flowing through the transistors.

ECL became extremely popular for use in sections of telecommunications equipment applications, generating logic families such as the 1000H and 10000H. ECL suffers from two major drawbacks however. First, ECL requires relatively high current as described above. Secondly, ECL relies on a negative power supply for operation. This can cause problems when interfacing to positive-supply-based devices residing in the rest of the system.

A new variant of ECL was developed to overcome the issues of interfacing with positive-voltage-based logic families such as TTL and CMOS. Positive Emitter Coupled Logic (PECL) has similar performance properties to ECL but operates from a positive voltage rail making it simpler to integrate into a digital system. Unfortunately PECL still suffers from the power issues of its predecessor.

Differential Systems

Single ended transmission is susceptible to noise. This can be overcome by overdriving the voltage but increases the power requirement, and because of the wider voltage swing, results in slower transition rates. Single ended transmission lines tend to attenuate the signal; again this is overcome by increasing the transmission voltage.

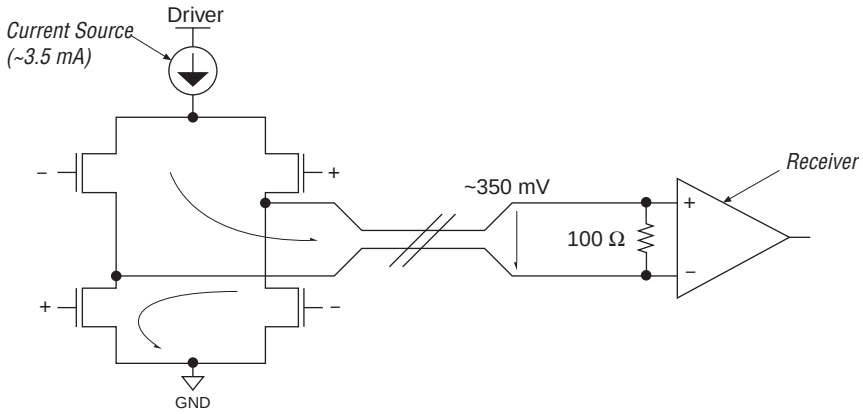
Differential buffers overcome these problems by transmitting a pair of signals of equal and opposite polarity for every bit sent. The receiver looks for the differences between the two signals and any noise common to both signals is rejected. Differential systems are less hampered by line attenuation because of their greater noise immunity and can therefore drive longer distances at higher data rates. For example the RS422-A differential standard supports transmissions at speeds of up to either 10Mbps for 10 meters or 100Kbps for one kilometer. Improvements in receiver technology have led to a reduction in transmission voltages by lowering the voltage swing to greatly increase transmission rates using techniques such as LVTTL. A number of low voltage differential I/O interfaces are now available and have been integrated into various interface standards.

Low Voltage Differential Signaling

The demands for high speed transmission data over chip-to-chip, board-to-board, and even longer distances have resulted in the development of the Low Voltage Differential Signaling (LVDS) I/O standard. Based on CMOS logic, LVDS features high speed with low noise generation, EMI resistance, and low power requirements. LVDS is used in high bandwidth data transfer applications, in particular backplane transceivers or clock distribution applications. A common reason for choosing LVDS is its low signal swing voltage of 350 mV, much lower than TTL, ECL and CMOS. This lower swing voltage presents added benefits over other alternatives:

- LVDS is a power efficient standard. AC power is low because the signal switch-over voltage is small, leading to low power dissipation per signal transition. DC power is also low because although each channel requires 3.5 mA, it is likely a single channel will be replacing a number of existing parallel channels.
- LVDS generates reduced levels of EMI. Device-generated EMI is dependent on frequency, output voltage swing and slew rate. Due to the low-voltage swing of the LVDS standard, the effects of EMI are much less than with CMOS, TTL, or other I/O standards.

Figure 1. LVDS current mode driver



LVDS is defined by two similar industry standards supporting different data rates:

- IEEE 1596.3 supports data rates up to 250 Mbps
- ANSI/TIA/EIA-644 recommends a higher data rate of up to 655 Mbps. This standard suggests a theoretical maximum of 1.923 Gbps (based on a lossless medium) and provides recommendations for fail-safe operation for the receiver under fault conditions.

Low Voltage Positive Emitter Coupled Logic

Low Voltage Positive Emitter Coupled Logic (LVPECL) is a low voltage differential version of PECL, commonly used in video graphics, data communication and telecommunication applications. LVPECL also forms the signaling standard for a number of protocols including Gigabit Ethernet and Fibre Channel. The LVPECL electrical specification is similar to LVDS, but operates with a larger differential voltage swing. LVPECL tends to be a little less power efficient than LVDS due to its ECL basis, however it can operate at frequencies above 1 Gbps due to its high speed switching characteristics.

Pseudo Current Mode Logic

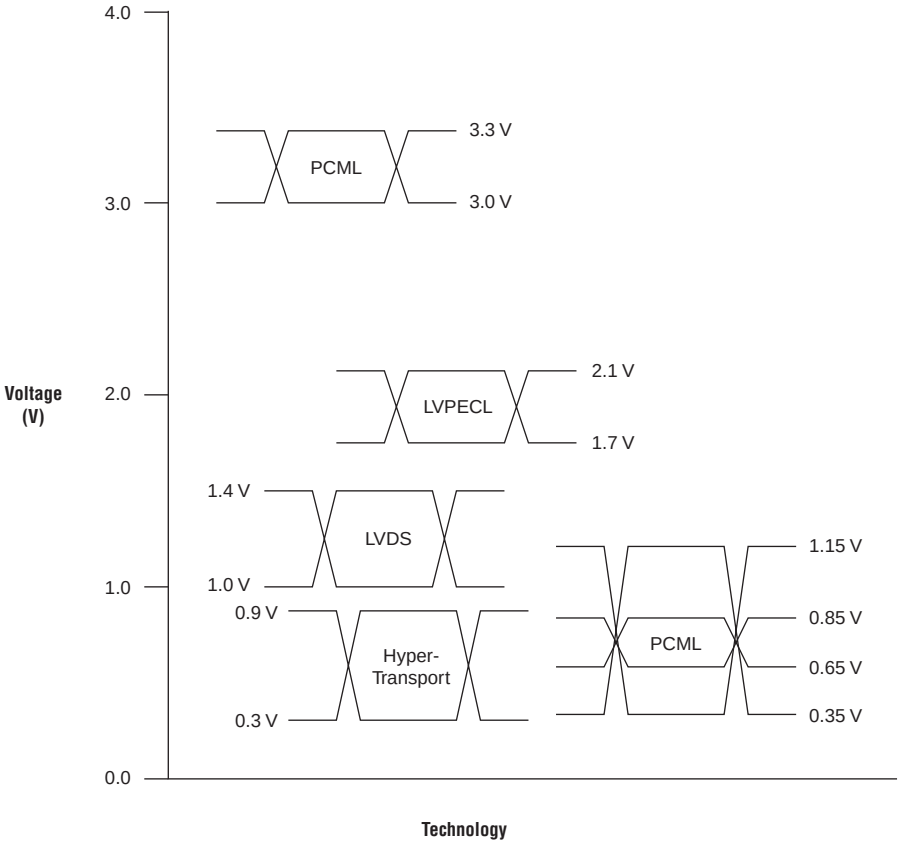
PCML is a high speed differential architecture derived from CML logic and is capable of speeds in excess of 2 Gbps. PCML operates from a 3.3-V power supply, higher than LVDS and LVPECL, but uses a lower swing voltage, allowing for faster switching. PCML has better power characteristics than its ECL ancestors, but has higher power consumption than other differential I/O standards not based on the current mode configuration. A later version of PCML has recently been specified to operate from a 1.5 V supply, facilitating even greater speeds. PCML has been widely adopted for use at data rates of 2.5 Gbps and above. PCML is used in a variety of applications including networking and data communications.

HyperTransport Technology I/O

HyperTransport Technology I/O, formerly known as Lightning Data Transport (LDT), has been developed to support next generation chip to chip interfaces for computer based equipment. Unsurprisingly, the standard is used as the signaling interface for the HyperTransport protocol. HyperTransport has been developed as an enhanced version of LVDS, employing a larger voltage swing. HyperTransport requires a 100 Ω differential terminator at the receiver buffer which when combined with the modifications to the signaling make it a more robust interface when used in an application with relaxed Printed Circuit Board (PCB) layout requirements.

The following figure (figure 2) below describes the various I/O standards discussed, showing both peak-to-peak voltage and differential swing.

Figure 2. Differential I/O Technologies



As memory devices become faster, new low-voltage I/O standards have been developed to provide high-speed memory support. Differential HSTL (Class I & II) operates with a 1.5 voltage range, with a 0.6 V peak to peak swing and requires a 50Ω termination resistor. HSTL can be used to interface to Quad Data Rate memory (QDR) or Synchronous SRAM. Differential SSTL-2 (Class I & II) operates with a 2.5 I/O voltage range, with a 0.6V peak to peak swing. SSTL is used to interface to DDR SDRAM.

SERDES

Increases in processor performance have resulted in changes in the methods for transferring data around the system. Chip-to-chip and backplane interfaces have traditionally been based on parallel bus interfaces, but ever-increasing data rates have made it more difficult to ensure data integrity when using these techniques. Data transfer has benefited from new signal technologies such as LVDS, which enable the movement of slow, parallel-form data to be converted into a serial form and transmitted using high speed serial links, simplifying board layout and design.

In order to multiplex the channels on a high speed transceiver, a serializer is used to multiply the parallel channels onto a single serial channel. This process is synchronized to a transmission clock source. The serial data stream is then transmitted, accompanied by the transmission clock source. The receiver circuitry receives the data, which is passed through a deserializer to return the data to the parallel form. The deserializer uses the transmitted clock to ensure the data is synchronized correctly.

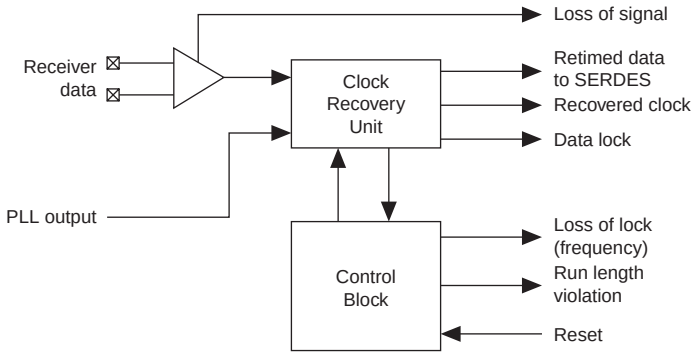
Initially the serializer/deserializer blocks—synchronous FIFO memory devices with mixed port widths—were provided as separate devices. But as transceivers reached higher levels of performance, it became necessary to embed the serializer-deserializer (SERDES) inside the transceiver block, thereby maintaining the data rate within the system.

As data rates continue to rise it becomes increasingly difficult to ensure synchronization between the clock and data. Board layout effects and connectors introduce skew between the clock and data. To guarantee data integrity it is important the system meets tight channel to channel skew and channel to clock skew. But this becomes virtually impossible with data rates in excess of a 1 Gbps using conventional clocking and PCB layout techniques, so new clocking methods must be introduced.

Clock Data Recovery

Clock data recovery (CDR) provides a technique of embedding the clock within the data to ensure data integrity. The transmission circuitry consists of a serializer and a synchronizer block. The synchronizer takes a clock source and uses this to serialize the data. This clock source is embedded into the data signal before transmission. The receiver consists of a clock recovery unit (CRU) and a deserializer. Data is fed via the receiver into the CRU, which takes the data stream and calculates the clock and phase from the transitions in the data. This clock can then be fed into the deserializer allowing for the data to be recovered in its original form. Figure 3 shows the block diagram of a clock data recovery system.

Figure 3. Clock Data Recovery Block Diagram



The CRU uses data transitions to determine the clock speed. Due to the lack of a separate clock signal, transitions from 1 to 0 and from 0 to 1 in the data itself must be used to infer a recovered clock. As such, CDR communication typically imposes a maximum run length, or a minimum transition density in order to retain lock. When a lull in data transfer occurs the CRU continues to run the clock at the rate of the previous data so it is possible when the new data arrives that it will not be synchronous to the clock. To overcome this, encoding techniques such as 8b10b are used to ensure that if a string of 1's or 0's is about to be transmitted, some bits are inverted so the run length is reduced. While these encoding techniques do add some degree of overhead to the data signal, that overhead is typically minimal when compared with the dramatic performance improvement enabled by CDR.

Data encoding is used within data transmission because of the ways it facilitates clock synchronization, DC balancing and error correction. The most used codes are schemes such as 5b6b, 8b10b or multiples including 64b66b.

Protocols such as 10 Gigabit Ethernet, Fibre Channel and Infiniband use the 8b10b encoding technique, which makes it viable for transceiver manufacturers to include the encoding block within the body of the transceiver. Block integration allows the data path to continue to run at high speed, as the functions no longer have to be processed in software.

The 8b10b scheme takes an 8-bit word and converts it into a 10-bit word by using a look up table approach. The new 10-bit word is selected to ensure the data will not produce a continual stream of '1's or '0's; the signal will not remain at the same level for more than 5 bits. Encoder words have been developed to have either an equal 5/5 or a 6/4 bit split to provide DC balancing. Additionally a look-back technique is employed to check for DC disparity to ensure a greater number of 1's or 0's have not been sent; therefore if a negatively balanced code is transmitted (6/4) the following code will be selected as positively balanced (4/6).

The 10-bit codes can support all 256 8-bit characters including disparity and in addition provide support for control words known as “comma” or “K” characters. Commas are used for transmitting control information such as idle characters, test data or data delimiters. K28.5 is used for alignment purposes with the code selected to ensure it cannot be transmitted when characters are merged within the bit stream. Table 1 shows a typical list of supported K or “comma” codes..

Table 1. 10-Bit Special K Codes Equivalent 8-Bit Codes

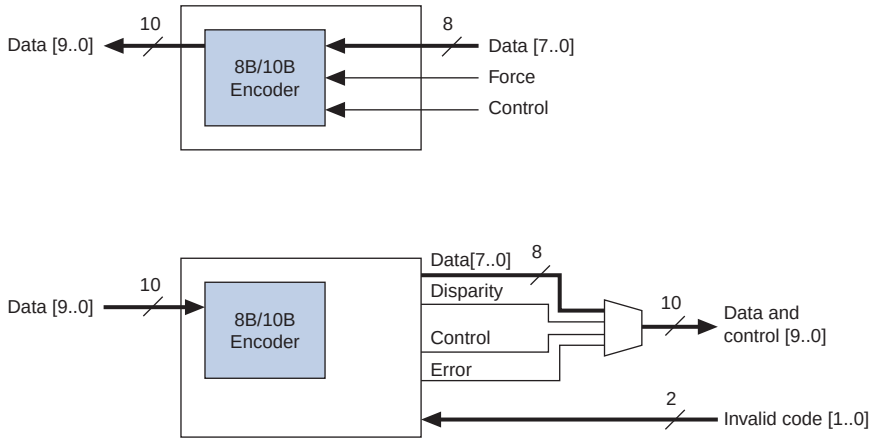
Code	8-bit equivalent
K28.0	8’b000_11100
K28.1	8’b001_11100
K28.2	8’b010_11100
K28.3	8’b011_11100
K28.4	8’b100_11100
K28.5 (1)	8’b101_11100
K28.6	8’b110_11100
K28.7	8’b111_11100
K23.7	8’b111_10111
K27.7	8’b111_11011
K29.7	8’b111_11101
K30.7	8’b111_11110

Note:

(1) K28.5 is a comma character used for alignment purposes, and to represent the IDLE code.

Figure 4 shows a block diagram of an 8b10b Encoder/Decoder.

Figure 4. 8b10b Encoder/Decoder Block Diagram



Another common encoding method is scrambling. Scrambling is employed by SONET/SDH based applications to encode data, ensure data clock synchronization, DC balancing and reduce EMI. There are a number of scrambling approaches available so it is not usual for these to be employed within the transceiver block and it is more common for these to be developed within logic or by using IP.

Data Alignment and Integrity

Once the clock has been recovered and the data has been passed through the SERDES, the data words must be realigned in order to properly identify the start of each byte. During the transmission phase, frame alignment characters such as commas are embedded into the serial stream to allow the receiver circuitry to detect alignment. The detector looks for the unique alignment codes within the data to determine the start of a valid string of data. The pattern detectors can offer some flexibility to allow for different encoding schemes. These are commas in 8b10b encoding, A1,A2 characters within the SONET/SDH protocol and even user-defined codes for custom backplane applications.

Once the pattern is detected, the incoming data can be realigned to the basic word boundaries to which it was transmitted. The pattern detector generates a signal to the word aligner to instruct it to realign the words to the boundary of the received pattern. Registers within the word aligner manipulate the data to the correct state depending on the protocol and byte width.

Multiple Channel Alignment

Many backplane protocol specifications now support simultaneous multiple channel transfer for high-speed data transmission, in order to achieve the required data rate. For example a protocol requiring 10-Gbps transmission across a backplane will often support 4 channels at 2.5 Gbps each to perform the task. In these applications it is important to ensure the data is realigned to its original state at the receiver. Channel alignment circuitry is often built into the receiver to support this function.

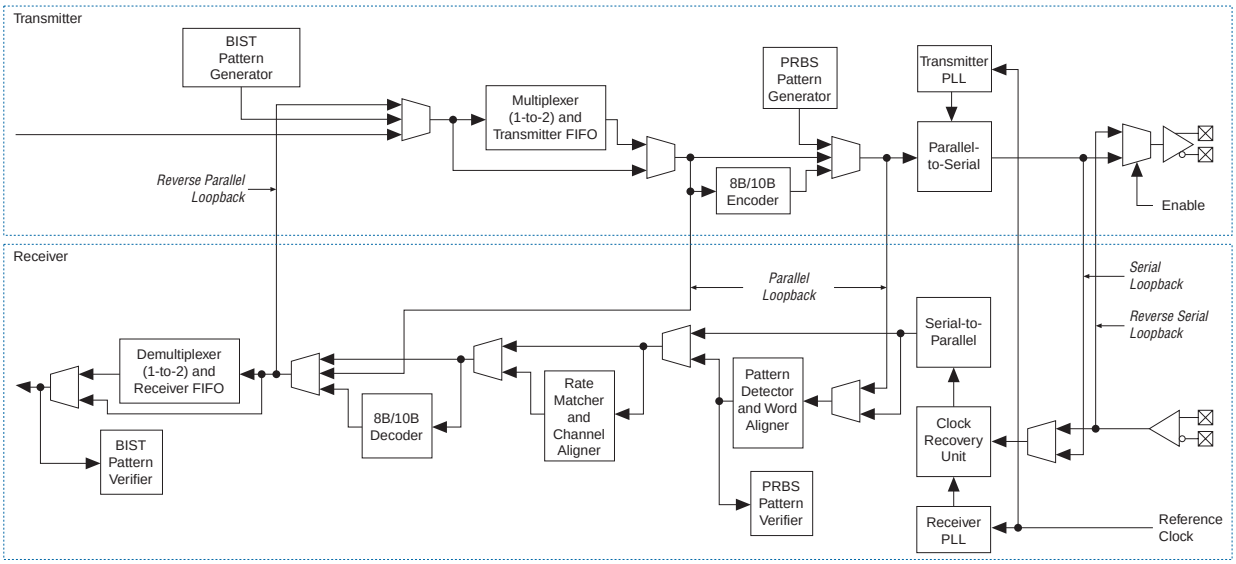
In multi-channel applications, the transmission circuitry will be required to insert code words simultaneously on all channels to show alignment. The receiver circuitry which consists of a control system and a FIFO will pass word aligned data into the FIFO from the receiver. The control system interrogates the received data for alignment characters. Once alignment characters have been received the control logic can synchronize all the channel FIFO's to a common clock, generally the receiver clock from the highest-order channel.

Rate Matching

High speed systems often rely on multiple clocks for operation. Often the receiver will have a receive clock and a master clock controlling the interface, while a core clock is used to control the data once it has left the transceiver section of the system. Rate matcher techniques are used to remove fluctuations between the two clocks that can cause data loss.

Many protocols call for transmitted data to be embedded with space or idle characters to help support rate matching. As the data arrives at rate matcher, data is passed into a FIFO. The rate matcher aligns the data to the core clock by adding or removing the space characters in order to allow the data to synchronize. The rate matcher will add some overhead to the data being transmitted but will result in a reduction in lost data and is useful when bridging between different interfaces. Figure 5 illustrates the transceiver blocks included in the Stratix GX architecture.

Figure 5. Stratix GX Transceiver Architecture

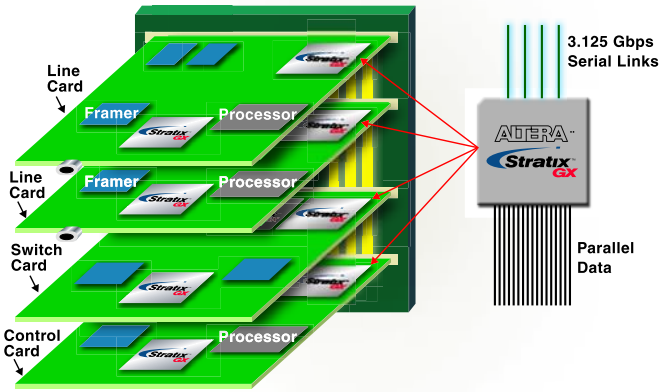


High-Speed Transceiver Applications

High-speed serial transceivers are now commonplace within the communications environment. Traditionally transceivers were limited to line-side applications, but as data bandwidth increased the need to move the data around the system as quickly as possible also increased. Transceivers are now being deployed across the entire system and are particularly useful for backplane applications and increasingly for chip-to-chip communication.

A typical communication system, as shown in Figure 6 below, is developed around a chassis of cards connected via a common backplane. These systems increasingly rely on high-speed transceiver technology throughout the system. The major function blocks within the unit include line cards, management and control cards and switch fabric cards.

Figure 6. Typical Transceiver Applications



Line-Side Applications

The line card is used to transfer system data to and from the outside world. Generally this data is passed through a line side transceiver and converted into or from optical data. Today's line cards support high-speed protocols such as Gigabit Ethernet, 10 Gigabit Ethernet, Fiber Channel, Infiniband and the 10-Gbps SONET/SDH based OC-192 interface, or 40-Gbps OC-768 interfaces. The line card connects to the rest of the system via a backplane. A stand-alone transceiver device is typically on the line-side but additional flexibility is sometimes required. As a result, FPGA implementations are also occasionally considered on the line side. In addition, the data rates generated by the line side have driven increasing requirements on the backplane.

Backplane Applications

System backplanes were traditionally developed around a parallel-style bus architecture. System standards such as VME, VXI and PCI evolved to support applications ranging from communications to industrial control. Backplane designers will rely on high-speed serial transceivers in an attempt to meet new data rates.

VME and VXI were initially the most popular backplane interfaces. This was due to what were, at the time, the wide 32-bit bus interfaces, deterministic interrupt schemes and most importantly the ability to support multiprocessor systems. These bus standards have more recently evolved to support 64-bit operation with VME64, capable of supporting data rates of up to 1 Gbps.

Recently the PCI based backplane has been adopted within the embedded arena. PCI is becoming popular due the abundance of off-the-shelf software solutions and the relative availability of experienced design engineers within the sector. Additionally, the embedded market requires an open system standard. The standard PC box interface did not provide the performance required for high speed applications so new standards have evolved based on higher data rates of 66 Mhz and larger data widths of 64 bits. The original form factor also proved unsuitable for many applications. Consequently new variants were spawned, like the Compact PCI used in chassis based applications and PMC modules, which can act as a mezzanine interface to host cards. The latest PCI-X V2.0 specification provides 64 bit DDR operation at 266 Mhz supporting a bandwidth of up to 4.2 gbytes/sec. However, as system bandwidth requirements increase, next generation architectures such as PCI Express must use high speed serial transceivers.

Even at the highest data rates, it is increasingly difficult to support the latest network protocols. A parallel signalling approach known as the XGMII interface is specified to transfer the data between MAC and PHY. In order to support

the full, data, control and clock interface, the specification calls for a 74 bit parallel interface operating at 312 Mhz. Although this is possible using a standard FR4 Printed Circuit Board, PCB layout issues such as skew crosstalk and jitter make it extremely difficult to produce a successful layout. Trace lengths with such an implementation would also be limited to around 6 cm, which is effectively useless if the information was to be transmitted across a backplane.

Designers are switching to high-speed serial based transceivers with built in CDR, SERDES and Encryption functionality to overcome these problems. The 10-Gigabit Ethernet specification specifies a XAUI interface, where the 12.5 gigabits of data is transmitted using four 3.125 Gbps transceivers capable of driving PCB trace lengths up to 40 inches, while making PCB design less complex.

The emergence of the backplane transceiver has caused a number of new protocols to evolve. These protocols are also becoming adopted for chip-to-chip and even box-to-box connection:

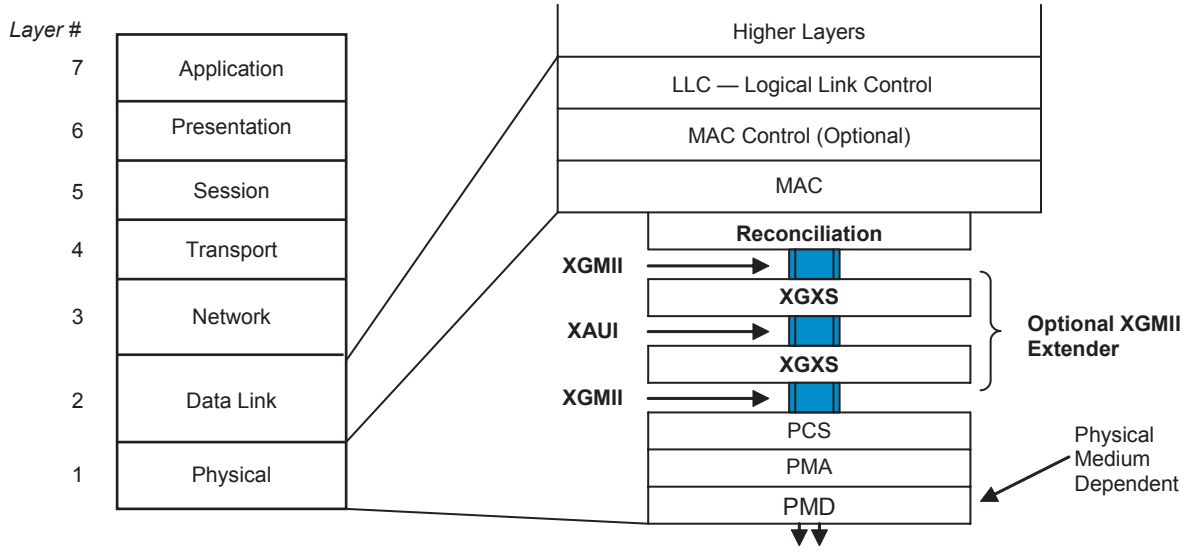
10-Gigabit Ethernet and Gigabit Ethernet

Ethernet architecture is used in local-area-network-based applications, where every computer system in the office is connected to the network using either a 10 Mbps or 100 Mbps link. Newer ethernet technology standards support data rates of 1 Gbps and 10 Gbps. This new performance not only helps to solve the ever increasing local Internet bandwidth but also make the technology viable further into the backbone. 10 Gigabit Ethernet has now seen adoption not only in the Metropolitan Area Network (MAN) but ever increasingly in the long haul networks, where it is used as an alternative to SDH and SONET/SDH, because the interface equipment is simplified, resulting in less network processing and thereby less bridging equipment.

XAUI

10 Gigabit Ethernet brings its own design challenges, especially in applications where 10 Gbps data must be transferred between the PHY and the MAC. 10 Gigabit Ethernet adopts the XGMII bus between MAC to PHY interface, but it is difficult to support the 74 bit interface operating at 312 Mhz over a long distances. A solution has been reached with the development of the XAUI interface (802.3 Clause 47), which can be introduced when the XGMII interface is inefficient to use. Figure 7 below shows the relative positioning of the interfaces within the OSI model. The XAUI interface provides four channels of 3.125 Gbps data per channel, capable of supporting 10 gigabits of data plus the 2.5 gigabits of overhead within the 10 Gigabit Ethernet payload.

Figure 7. Interface Positions Relative to the OSI Model



The XAUI interface employs high-speed serial transceivers requiring CDR to support the high data rates required for the interface. The 8b10b encryption techniques, generally used by all higher speed ethernet standards, ensure error free data, clock synchronization and DC balancing. XAUI is a multi-channel interface primarily used for PCB and board-to-board connectivity and contains specific provisions for rate matching and channel alignment circuitry. The XAUI interface will also provide support for SONET/SDH STS-192/STM-64 traffic at those rates, to support its use in MAN and long haul applications.

Infiniband

Infiniband was initially developed by a number of leading computer manufacturers to provide a standard interface for storage and server applications. Infiniband operates as a point-to-point interface and is seen as either a board-to-board or box-to-box connection. Infiniband is capable of data rates of up to 30 Gbps using 2.5 Gbps serial transceivers utilizing CDR in either 1x 4x or 12x bus configurations. Infiniband utilizes SERDES technology, and 8b10b encoding/decoding techniques, relying on rate matching and channel alignment to ensure data packet synchronization.

Serial Rapid I/O

Rapid I/O was developed by a number of major telecommunications manufacturers as a system interface for chip-to-chip and backplane interconnect. The initial specification called for a parallel implementation because the interface was developed as the next-generation PCI interface including multiprocessing extensions. Serial Rapid I/O supports data rates of 1.25, 2.5, and 3.125 Gbps and supports various channel widths.

As with other backplane interfaces discussed in this section, Serial Rapid I/O requires many of the high-speed serial transceiver building blocks, including CDR, SERDES, and 8b/10b encoding. 4LP mode also requires rate matching and channel alignment to ensure the data is kept sequential.

Fibre Channel

Fibre Channel has become very popular in storage-area networks by marrying the best of the network interface with the best of the system interfaces. Storage networks have generally been built around parallel bus structures such as SCSI, but limitations in existing parallel transceiver technologies have stunted data rates and provided inflexibility over line lengths.

High performance transceiver technology has helped to overcome these issues. Fibre Channel provides for two standard data rates, 1-Gbps and 2.125-Gbps, with a 10-Gbps standard currently under development. Fibre Channel can be employed using a number of different topologies but at high data rates, a point to point interface is generally used. Fibre channel uses combined clock/data signaling and 8b10b encoding at the physical level to ensure successful transmission at high frequency. At the data level, Fibre Channel supports both the SCSI and IP protocols to allow interfacing with legacy storage equipment.

PCI Express (also known as 3GIO and Arapahoe)

PCI-Express (formerly known as 3GIO) is being developed to provide a high speed backplane alternative to PCI. As network processors have increased to speeds in excess of 10Gbps, and with the introduction of high speed peripheral and graphics interfaces, the parallel-architecture PCI bus and derivatives have been less viable. The PCI Express architecture makes use of high speed transceiver technology to overcome the speed and layout limitations of the original PCI specification.

PCI Express is a scalable point to point architecture based on 3.125 Gbps transceivers and will operate over x1, x2, x4, x8, x12, x16 or x32 lane widths using split-byte technology. The lane width is not prohibitive as a negotiation stage is held between the master and the peripheral to determine the lane width and frequency when the link is first established. Data is transmitted with an embedded clock and uses 8b10b encoding methods to ensure clock integrity

and DC balance. The receiver requires detection and alignment technology to ensure the data can be reassembled in the original form.

PCI Express based systems will consist of a number of switches and bridges allowing more established architectures to co-exist. The software layer will remain compatible with the existing PCI specifications to limit changes to operating systems while maintaining the robust configuration and initialization schemes used today.

Table 2 shows the protocol block requirements of the standards discussed in this section.

Table 2. Dedicated protocol transceiver block requirements

Transceiver Elements	Standards					
	10 Gigabit Ethernet XAUI	1Gigabit Ethernet	Serial RapidIO	Fibre Channel	InfiniBand	PCI Express
CDR	✓	✓	✓	✓	✓	✓
SERDES	✓	✓	✓	✓	✓	✓
Pattern detector	✓	✓	✓	✓	✓	✓
Word aligner	✓	✓	✓	✓	✓	✓
8B/10B encoder/ decoder	✓	✓	✓	✓	✓	✓
Channel aligner	✓					
Rate matcher	✓	✓				
Synchronizer	✓	✓	✓	✓	✓	✓

Chip-to-Chip

Although a relatively recent innovation, the use of high-speed serial I/O for chip-to-chip interfacing is becoming increasingly adopted. Many applications, such as cross switches and bridges, need to pass data between chips but in order to meet the performance specification speed must be maintained. Traditionally this was achieved using parallel I/O, but the same transmission effects exist as with backplane interfaces, making the switch to high speed serial I/O necessary.

Many of the protocols adopted for backplane interfaces are also used for chip-to-chip interfacing. XAUI, Infiniband and Serial Rapid I/O can all be useful for chip-to-chip communications.

SPI-5

As SONET/SDH line rates increase to 40 Gbps (OC-768), it is sometimes necessary to split functionality between devices. The data rate between devices must be maintained however. SPI-5 specifies an interface between the physical layer and link layer to ensure data rates.

SPI–5 is a 16 channel high serial interface operating between 2.5 Gbps and 3.125 Gbps per channel, depending on payload. Data is transmitted in bursts based on 32-byte boundaries, which are interleaved onto the bus. Data alignment is provided by transmitting training words across the bus to ensure all 16 lines can be synchronized. The transceiver incorporates a pattern detector to identify the training words and ensure the channels are correctly aligned.

SPI-5 uses a common X11 + X9 + 1 LFSR technique for encoding, to aid clock synchronization and DC balancing. This technique tends not to be part of the transceiver as it is generally simple to build within logic.

SFI-5

SFI-5 is an emerging standard describing the electrical interface between the SERDES in the SONET/SDH transceiver and the SONET/SDH framer. The transceiver side of the interface is provided using seventeen 3.125 Gbps channels. Each channel feeds a SERDES converting the channel into sixteen channels of 195 Mbps. The deserialized channels may then be passed to other functions in the system using source synchronous methods.

The Advantage of Including High-Speed Transceivers within FPGA Technology

The evolution of the transceiver has been dramatic. Initially, high-speed I/O interfaces were built around separate ASSP devices, each providing a function such as LVDS transceiver, CRU, SERDES, or encode/decode functionality. However as data performance increased, these functions needed to merge to ensure performance throughout the data path and reduce board size. For instance, it is now common to find a XAUI solution provided within a single MAC/PHY device.

A more effective solution for high volume applications has seen the inclusion of transceiver technology within Application Specific Integrated Circuits (ASICs). Many ASIC manufacturers quickly adopted the transceiver technology by adding I/O resources to support the various signaling interfaces and adding other blocks such as CDR depending on protocol requirements. The remainder of the ASIC could then provide the rest of the system architecture, allowing functions such as network processing to be developed in the base system without the performance degradation associated with moving the data off-chip.

This system-on-a-chip approach using built in transceivers has many advantages. The ability to develop most of the system on a single device reduces the device count, thus reducing PCB space and overall system power consumption. PCB layout is also easier because it is no longer necessary to route as many high-speed busses around the system.

The rapid reduction in silicon die size and increased costs of manufacture have caused ASIC costs to become prohibitive for all but the highest volume designs. Reduced-size die technology towards 0.09 um has caused a major increase in NRE charges, while larger wafers have meant an increase in minimum-order quantities. A further cost factor is the lengthy design phase which can cause the product to be late or miss the market altogether. This is particularly true with embedded transceiver technology where design verification and manufacturing times are increased due to the complexity of integrating high-speed drivers onto the same silicon as digital logic. This integration often leads to more than one revision of the silicon.

ASICs also suffer from the effects of inflexibility. Many protocols are still in an evolutionary state and subtle architectural changes either within the system logic are often required. The fixed nature of an ASIC makes it extremely difficult to respond to these changes, resulting in a costly redevelopment process and the prospect of obsolete products.

The development of transceiver technology within a programmable logic device has provided a powerful alternative for high-speed system design. For some time, the development of this high-speed architecture enabled Altera to provide a source synchronous solution at speeds up to 1 Gbps, as in the APEX II architecture. More recently the inclusion of more of the transceiver building blocks into the body of the I/O has lead to the release of FPGAs with fully integrated high speed transceiver I/O functionality. Altera's Mercury device was the industry's first high-speed device to integrate CDR and SERDES functionality within the transceiver. The rest of the FPGA could then be used to create the remaining transceiver blocks and system interfacing. Mercury devices are capable of data rates up to 1.25 Gbps per channel across a maximum of 18 channels.

Stratix GX is the latest Altera device family to include high-speed transceiver technology. It supports multiple channels of data at 3.125 Gbps per channel. with the largest device capable of supporting over 40 Gbps of data, plus

overhead, across 20 channels. The Stratix GX transceiver has been developed to support new high-speed communication standards such as XAUI, Infiniband and SONET/SDH and provides the CDR, SERDES, encoding, alignment and rate matching circuitry required within the transceiver block. The Stratix GX device can fully support both the physical coding sublayer (PCS) and Physical Media Attachment (PMA) layer of the Open System Interconnect (OSI) model for XAUI without the need for external logic, leaving the rest of the device free for other system requirements. Each block is configurable, allowing the same device to support a number of different standards.

In addition, Stratix GX includes Dynamic Phase Alignment (DPA) for source synchronous data. DPA can be used for applications where slew exists between clock and data signals or for standards where CDR functionality is not permitted. The DPA circuitry divides the receive clock into 8 phases and aligns the data stream to the closest phase of the clock, thus eliminating the effect of board skew. DPA is effective up to data rates of 1 Gbps, and provides a complementary solution to help enable access to the high bandwidth of data offered by the Stratix GX transceivers.

Some protocol standards call for alternative functions to be added to the blocks provided within the transceiver section. Stratix GX can be programmed to bypass these functions, allowing the engineer to replace the block with either off-the-shelf or in-house intellectual property, should a standard change or a new standard emerge. For example if a system requires a scrambler-based encoding scheme rather than a 8b10b coding, the encoder block can be bypassed allowing the user to develop the scrambler architecture in the user logic. If the scrambler code changed, the logic could easily be altered to meet the new code. The FPGA architecture also allows the user to develop propriety backplane interfaces.

Stratix GX provides many advantages beyond the I/O boundary. The device architecture has been developed to operate within the application data path. By building the system blocks within the FPGA it is possible to maintain data rates throughout the system. The architecture is based on the high performance Stratix family and includes three different memory sizes suitable for various networking tasks. For example the large M-RAM blocks can be used, as FIFOs to support queuing packets while the M4K blocks are ideal for holding header information. Additionally network-processing tasks can be handled within the FPGA by including Nios® soft processor cores within the device to support the system.

The design risks associated with using FPGAs like Stratix GX are also reduced. FPGA development times are short and there are no NRE costs or high minimum order quantities. The high speed I/O technology embedded into the FPGA is also assured. The product has been designed for the mass market, consequently the I/O has been fully tested and characterized by the manufacturer, providing minimal risk to the customer.

However in high-speed data application it is still not always possible to provide a solution in a single device. In applications of extremely high data rates, for example SFI-5 where data streams of 40 Gbps are supported, or in high speed bridging applications ASSPs can co-exist with high-speed transceivers to provide a complete solution. In bridging applications, data arriving at the ASSP transceiver is split into several channels of high-speed I/O. This data is passed into the high-speed transceiver based FPGA using channel-aligned transceivers. Once in the FPGA, the data can be manipulated and transformed into a different protocols before transmitting to the second ASSP transceiver, again using multiple channels of high-speed I/O. The data can then be transmitted to the outside world in a different protocol from the input ASSP. FPGA technology interoperability is not restricted to the ASSP. High speed FPGAs may also interface with ASIC technologies or even other programmable high-speed transceiver based devices depending on system requirements.

High speed transceiver based FPGA architecture can sit anywhere within the data path, either as a stand-alone block or as a part of a combined-system approach. FPGAs are particularly suited for roles as replacements for a multi-chip solution or a high-maintenance low-flexibility ASIC development, particularly as network protocols continue to evolve.

Conclusion

The need for higher data rates has caused an explosion in transceiver technology. The development of programmable logic with integrated high-speed transceivers has many advantages in applications such as backplane and chip-to-chip interfacing because it is possible not only to develop the transceiver but also other key design blocks required to maintain the signal data rate. The use of a single chip solution also reduces system design headaches including the PCB layout issues related to multiple high speed devices, power requirements and support for continually evolving standards.

Altera's new Stratix GX family is ideally suited to applications requiring high speed transceiver technology. The transceiver blocks on these devices, merged with flexibility of FPGA technology including fast time to market make Stratix GX devices the ideal choice, whether as a stand alone transceiver solution or when interfacing with a 3rd party transceiver product.

101 Innovation Drive
San Jose, CA 95134

(408) 544-7000

http://www.altera.com