The ispGDX® 2 Family

White Paper

October 2002

Overview

Many designs require the implementation of switching and multiplexing functions or the translation of signals from one electrical interface to another. Lattice Semiconductor developed ispGDX, In-System Programmable Generic Digital X-point, devices to address these requirements without the cost burden of additional logic associated with implementing these functions in programmable logic devices such as CPLDs and FPGAs. Through ISP™ and JTAG boundary scan test ispGDX devices also provide flexibility for design changes and testability at key system interfaces.

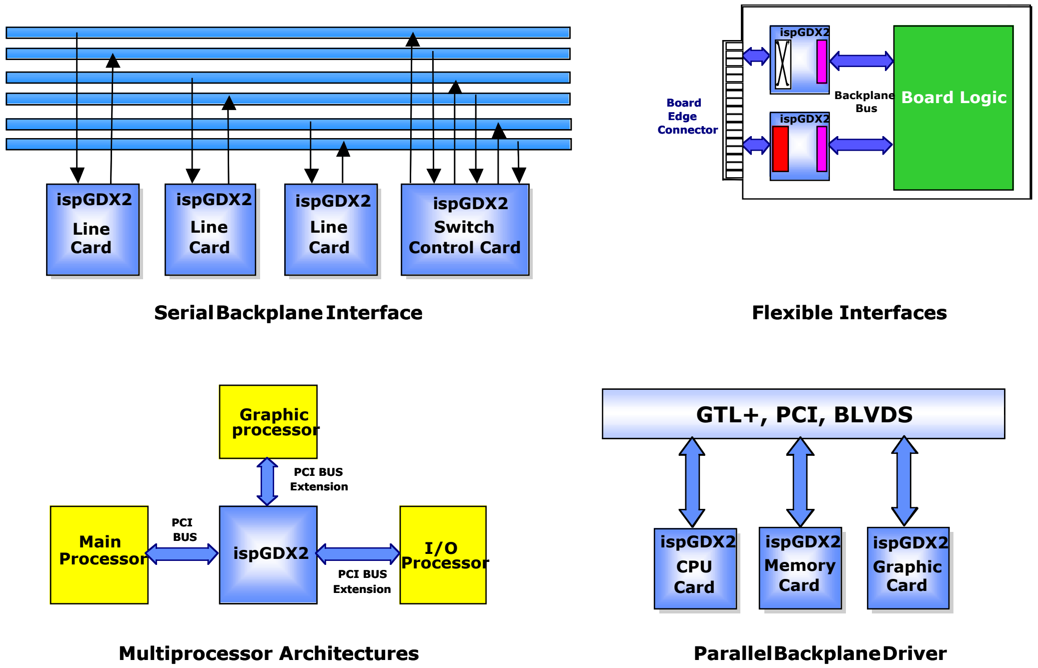
The ispGDX2 devices expand on the original ispGDX concept by adding several new capabilities. New programmable control logic blocks complement the ispGDX's switching functionality. Interfacing capability is enhanced with sysIO™ interfaces that provide support for a wide variety of interface types such as LVDS, Bus-LVDS, LVPECL, LVCMOS, SSTL, and HSTL. The addition of up to 16 sysHSI™ SERDES channels provide the capability to implement multiple serial interfaces that run at up to 850Mbps. Up to four sysCLOCK™ PLLs allow designers to adjust timing and synthesize new clock frequencies.

Applications

The ispGDX2 devices are suitable for many applications including the implementation of switched backplanes, low-cost serial links, standard translation, bus multiplexing, and the consolidation of drivers, transceivers, and SERDES. The devices provide the flexibility and time-to-market associated with programmability along with the benefits of JTAG testability to these applications. Figure 1 shows a graphical representation of some of these applications.

Because the ispGDX2 devices are tailored to switching and interfacing they represent a solution that is considerably more cost effective in these applications than other programmable solutions.

Figure 1 – ispGDX2 Applications



Serial Backplane Interface

Flexible Interfaces

Multiprocessor Architectures

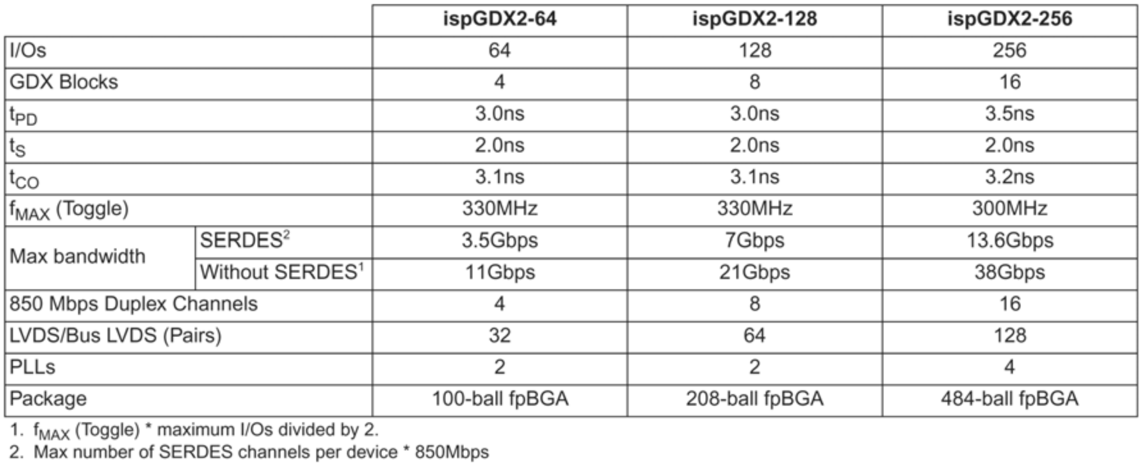
Parallel Backplane Driver

Family

Three devices make up the ispGDX2 family, spanning from 64 to 256 I/Os. (See table 1.) The devices provide high-performance switching with pin-to-pin delays as fast as 3.0ns and operating frequencies up to 330MHz. This performance allows the family to provide a maximum throughput of between 11 and 38Gbps depending on the device size.

Devices in the family can operate at 3.3V, 2.5V or 1.8V core voltages and can be programmed in-system via an IEEE 1149.1 interface that is compliant with the IEEE 1532 standard. Voltages required for the I/O buffers are independent of the core voltage supply. This further enhances the flexibility of this family in system designs.

Table 1 – ispGDX2 Family Members



		ispGDX2-64	ispGDX2-128	ispGDX2-256
I/Os		64	128	256
GDX Blocks		4	8	16
t_{PD}		3.0ns	3.0ns	3.5ns
t_S		2.0ns	2.0ns	2.0ns
t_{CO}		3.1ns	3.1ns	3.2ns
f_{MAX} (Toggle)		330MHz	330MHz	300MHz
Max bandwidth	SERDES[2]	3.5Gbps	7Gbps	13.6Gbps
	Without SERDES[1]	11Gbps	21Gbps	38Gbps
850 Mbps Duplex Channels		4	8	16
LVDS/Bus LVDS (Pairs)		32	64	128
PLLs		2	2	4
Package		100-ball fpBGA	208-ball fpBGA	484-ball fpBGA

1. f_{MAX} (Toggle) * maximum I/Os divided by 2.
2. Max number of SERDES channels per device * 850Mbps

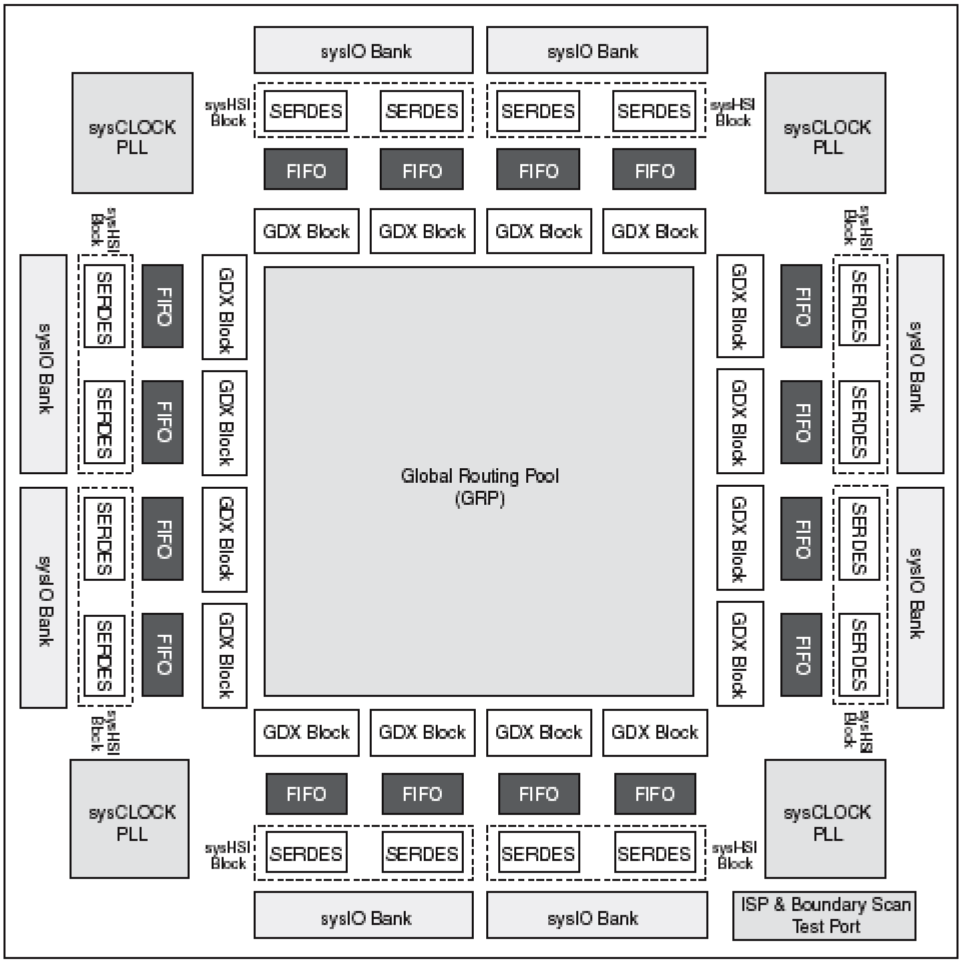
Architecture

The ispGDX2 devices consist of GDX Blocks interconnected by a Global Routing Pool (GRP). Signals interface with the external system via sysIO banks. In addition, each GDX Block is associated with a FIFO and a sysHSI Block to facilitate the transfer of data on- and off-chip. Figure 2 shows the ispGDX2 block diagram.

Global Routing Pool (GRP)

The ispGDX2 architecture is organized into GDX Blocks, which are connected via a Global Routing Pool. The innovative GRP is optimized for routability, flexibility and speed. All the signals enter via the GDX Block. The block supplies these either directly or in registered form to the GRP. The GRP routes the signals to different blocks, and provides separate data and control routing. The data path is optimized to achieve faster speed and routing flexibility for nibble-oriented signals. The control routing is optimized to provide high-speed bit oriented routing of control signals.

Figure 2 – ispGDX2 Block Diagram



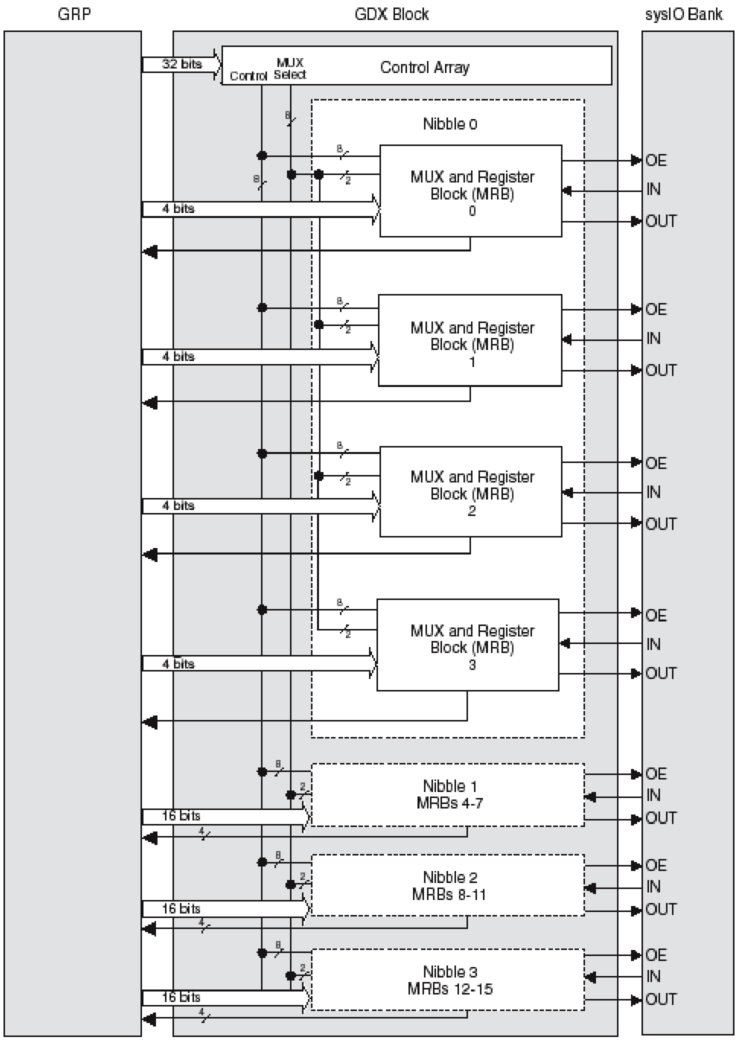
GDX Block

The blocks are organized in a “block” (nibble) manner, with each GDX Block providing data flow and control logic for 16 I/O buffers. The data flow is organized as four nibbles, each nibble containing four Multiplexer Register Blocks (MRBs). Data for the MRBs is provided from 64 lines from the GRP. Figure 3 illustrates the groups of signals going into and out of a GDX Block. Control signals for the MRBs are provided from the Control Array. The Control Array receives the 32 signals from the GRP and generates 16 control signals.

MUX and Register Block (MRB)

Every MRB Block has a 4:1 MUX (I/O MUX) and a set of three registers which are connected to the I/O buffers, FIFO and sysHSI Blocks. Multiple MRBs can be combined to form large multiplexers up to 188 to 1. Each of the three registers in the MRB can be configured as edge-triggered D-type flip-flop or as a level sensitive latch. One register operates on the input data, the other output data and the last register synchronizes the output enable function. The input and output data signals can bypass each of their registers.

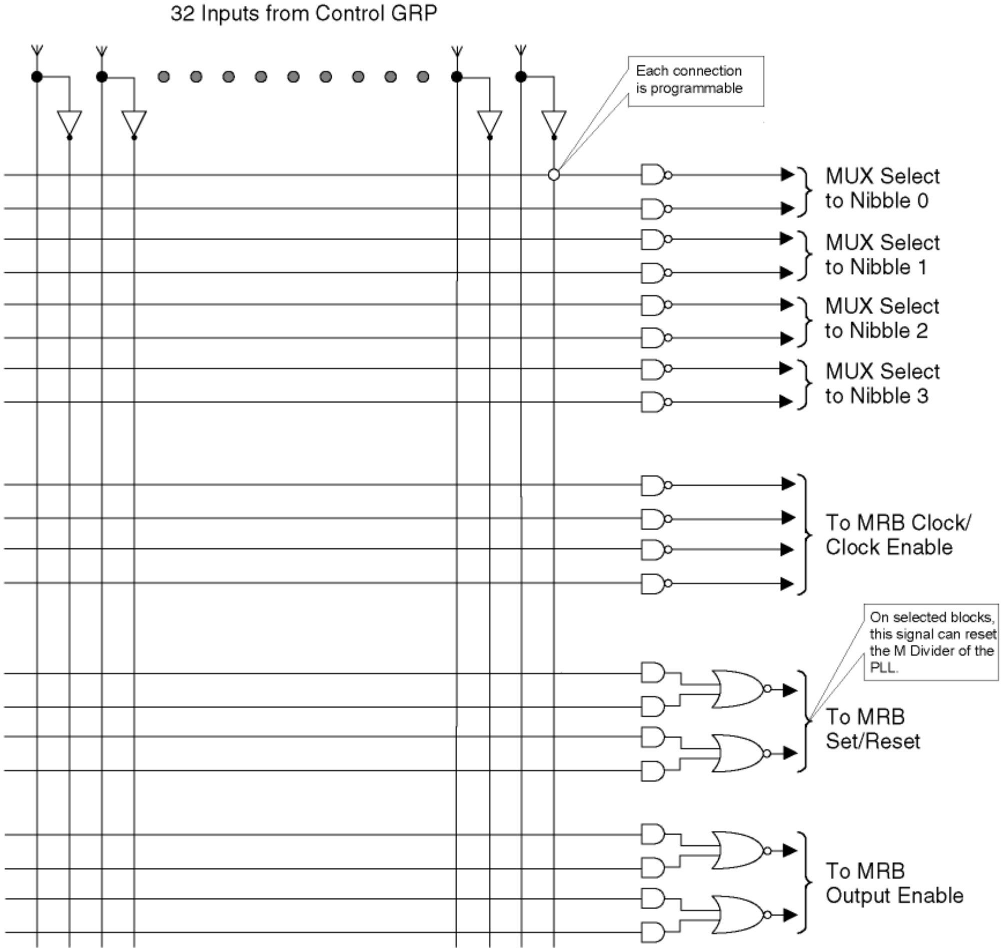
Figure 3 – GDX Block



Control Array

The control array generates control signals for the 16 MRBs within a GDX Block. The true and complement forms of 32 inputs from the GRP are available in the control array. The 20 NAND terms can use any or all of these inputs to form the control array outputs. Two NAND terms are combined with an OR terms to form Set/Reset and OE signals. Figure 4 illustrates the control array.

Figure 4 – Control Array


32 Inputs from Control GRP
Each connection is programmable
MUX Select to Nibble 0
MUX Select to Nibble 1
MUX Select to Nibble 2
MUX Select to Nibble 3
To MRB Clock/ Clock Enable
On selected blocks, this signal can reset the M Divider of the PLL.
To MRB Set/Reset
To MRB Output Enable

sysIO Banks

The inputs and outputs of ispGDX2 devices are divided into eight sysIO banks, where each bank is capable of supporting different I/O standards. The number of I/Os per bank is 32, 16 and 8 for the 256-, 128- and 64-I/O devices respectively. Each sysIO bank has its own I/O supply voltage (V_{CCO}) and reference voltage (V_{REF}), allowing each bank complete independence from the other banks. Each I/O within a bank can be individually configured to any standard consistent with the V_{CCO} and V_{REF} settings. The I/O of the ispGDX2 devices contain a programmable strength and slew rate tri-state output buffer, a programmable input buffer, a programmable pull-up resistor, a programmable pull-down resistor and a programmable buskeeper latch. These programmable capabilities allow the support of a wide range of I/O standards.

There are three classes of I/O interface standards implemented in the ispGDX2 devices. The first is the non-terminated, single-ended interface; it includes the 3.3V LVTTL standard along with the 1.8V, 2.5V and 3.3V LVCMOS interface standards. The slew rate and strength of these output buffers can be controlled individually. Additionally, PCI 3.3, PCI-X and AGP-1X are all subsets of this interface type. The second interface class implemented is the terminated, single-ended interface standard. This group of interfaces includes different versions of SSTL and HSTL interfaces along with CTT and GTL+. Use of these I/O interfaces requires an additional V_{REF}

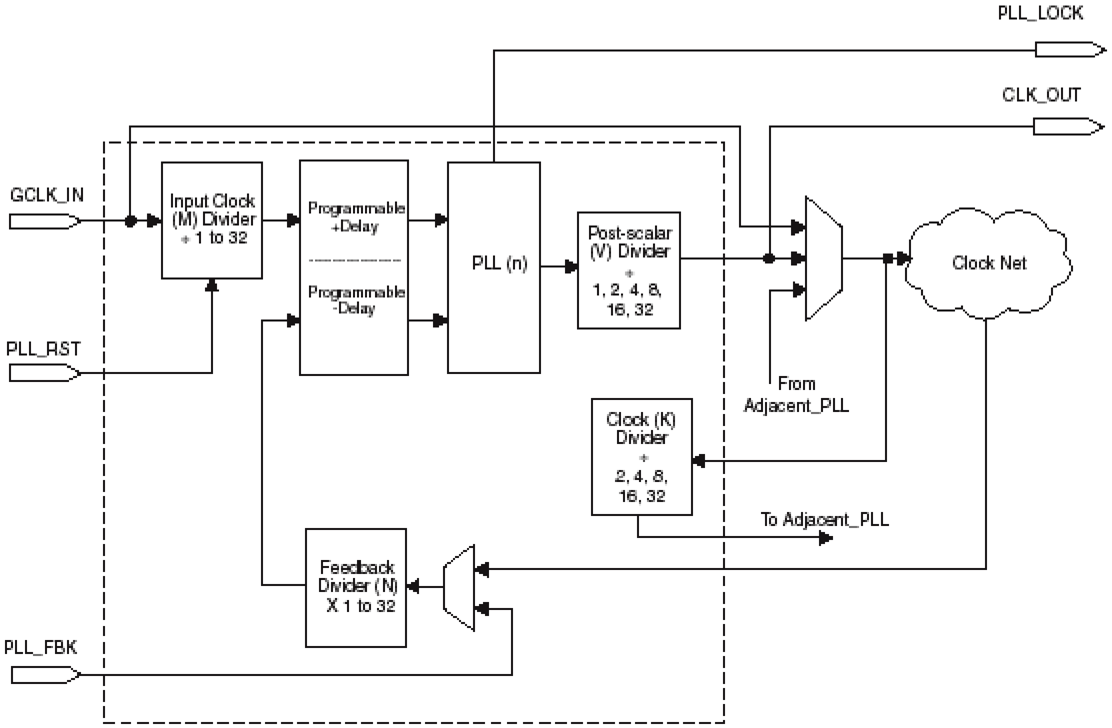
signal. At the system level, a termination voltage, VTT, is also required. Typically, an output will be terminated to VTT at the receiving end of the transmission line it is driving. The final types of interfaces implemented are the differential standards LVPECL, LVDS and Bus LVDS.

sysCLOCK PLL

The sysCLOCK PLL circuitry consists of Phase-Lock Loops (PLLs) along the various dividers and reset and feedback signals associated with the PLLs. This feature gives the user the ability to synthesize clock frequencies and generate multiple clock signals for routing within the device. Furthermore, it can generate clock signals that are deskewed either at the board level or the device level. Figure 5 shows the ispGDX2 PLL block diagram.

Each PLL has a set of PLL_RST, PLL_FBK and PLL_LOCK signals. In order to facilitate the multiply and divide capabilities of the PLL, each PLL has associated dividers. The M divider is used to divide the clock signal, while the N divider is used to multiply the clock signal. The K divider is used to provide a divided clock frequency of the adjacent PLL. This output can be routed to the global clock net. The V divider is used to provide lower frequency output clocks, while maintaining a stable, high frequency output from the PLL's VCO circuit. The PLL also has a delay feature that allows the output clock to be advanced or delayed to improve set-up and clock-to-out times for better performance.

Figure 5 – sysCLOCK PLL

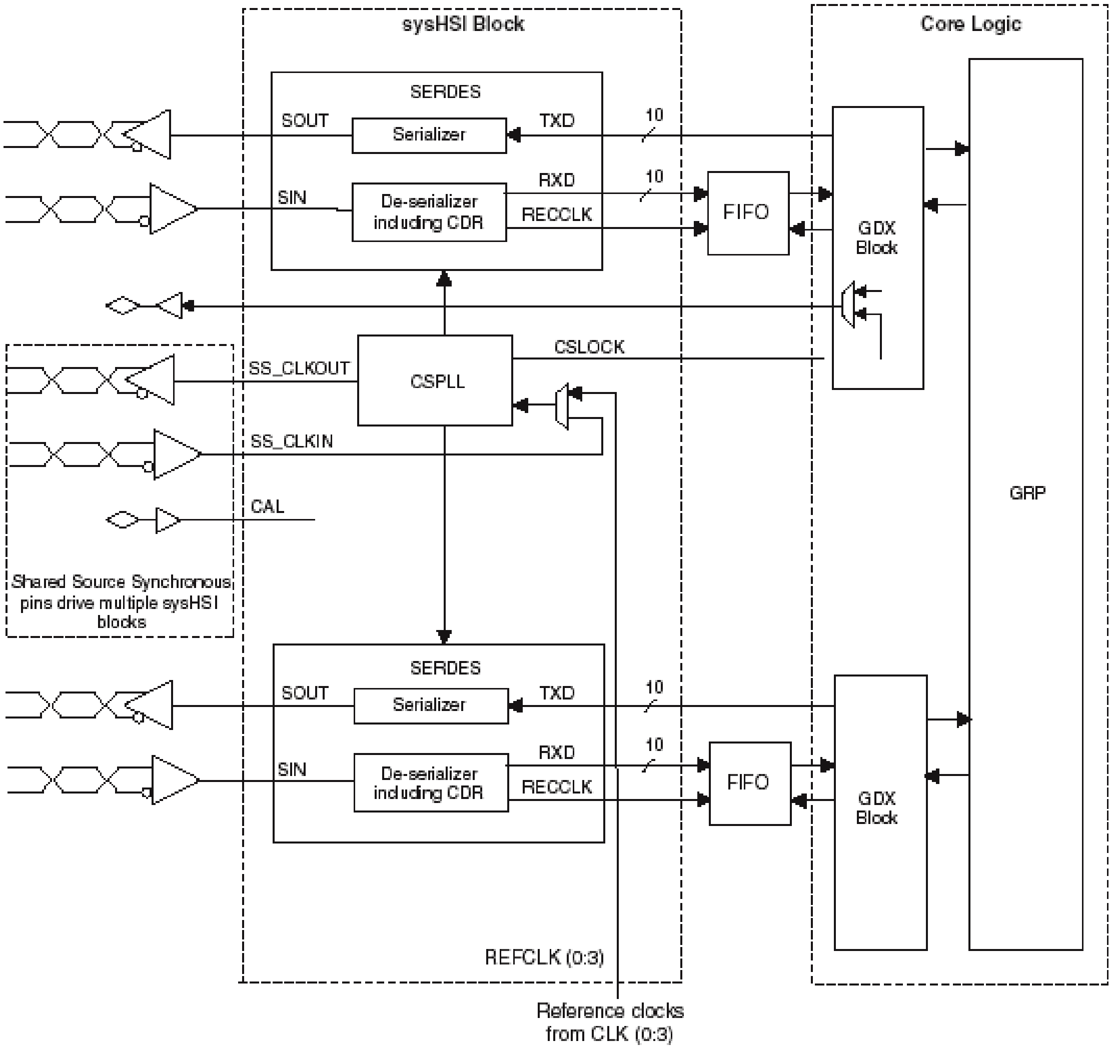

PLL_LOCK
CLK_OUT
GCLK_IN
Input Clock (M) Divider ÷ 1 to 32
Programmable +Delay
Programmable -Delay
PLL (n)
Post-scalar (V) Divider ÷ 1, 2, 4, 8, 16, 32
Clock Net
PLL_RST
From Adjacent_PLL
Clock (K) Divider ÷ 2, 4, 8, 16, 32
To Adjacent_PLL
Feedback Divider (N) X 1 to 32
PLL_FBK

High Speed Serial Interface Block (sysHSI Block)

The High Speed Serial Interface (sysHSI) allows high speed serial data transfer over a pair of LVDS I/O. The ispGDX2 devices have multiple sysHSI Blocks. Each sysHSI Block has two SERDES blocks which contain two main sub-blocks, Transmitter (with a serializer) and Receiver (with a deserializer) including Clock/Data Recovery Circuit (CDR). Each SERDES can be used as a full duplex channel. The two SERDES in a given sysHSI Block share a common clock and must operate at the same nominal frequency. Figure 6 shows the sysHSI Block.

Device features support two data coding modes: 10B/12B and 8B/10B. The encoding and decoding of the 10B/12B standard are performed within the device in dedicated logic. For the 8B/10B standard, the symbol boundaries are aligned internally but the encoding and decoding are performed outside the device. Each SERDES block receives a single high speed serial data input stream (with embedded clock) from an input, and provide a low speed 10-bit wide data stream and a recovered clock to the device. For transmitting, the SERDES converts a 10-bit wide low-speed data stream to a single high-speed data stream with embedded clock for output. Additionally, multiple sysHSI Blocks can be grouped together to form a source synchronous interface of between 1-8 channels.

Figure 6 – sysHSI Block

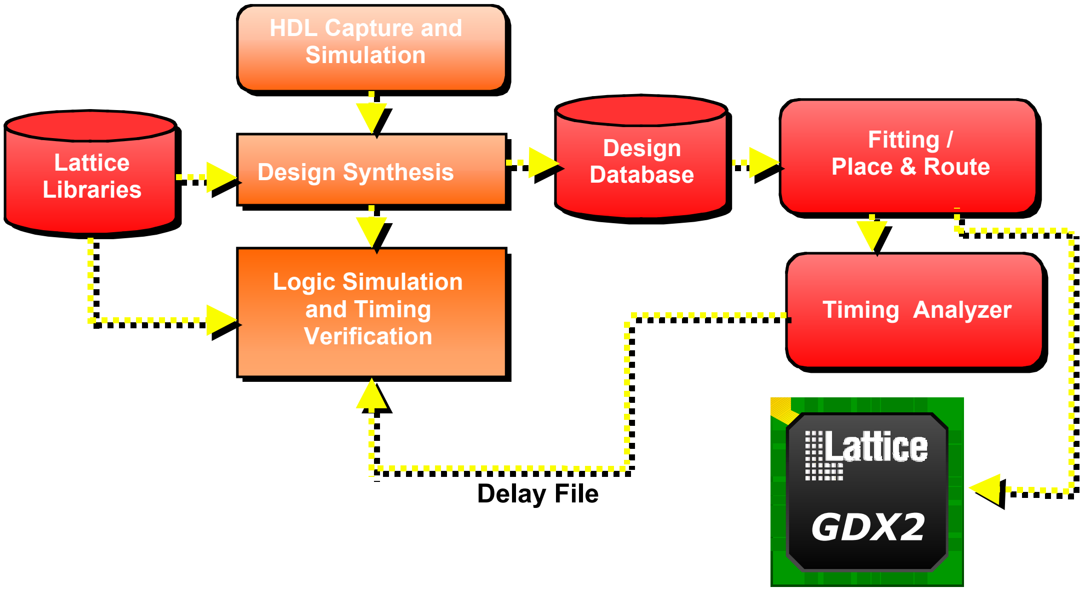


Design Environment – *ispLEVER™*

Lattice's new ispLEVER design tool supports ispGDX device design. ispLEVER has been designed as Lattice's design platform for the future. Building on the foundations of past design tools, ispLEVER includes fitting and usability enhancements that speed design and improve achievable performance. The ispLEVER design tools support a new Constraints Manager for easy sysIO and sysCLOCK configuration and pin assignment, enhancements to Lattice's Performance Analyst^{TM}, static timing analyzer, as well as other ease-of-use refinements, such as HTML Reporting and Navigation and automatic web-based ispUPDATE^{TM}. Every ispLEVER product also includes Lattice's powerful In-System Programming tool, the ispVM^{TM} System.

The ispLEVER tool supports design in all of Lattice's digital devices whether CPLDs, FPGAs, or GDXs. This means users only have to learn one user interface no matter which product they chose to use for implementing a particular design. ispLEVER design tools also integrate leading 3rd party CAE vendor tools for synthesis and RTL and timing simulation from Mentor Graphics. / Exemplar and Synplicity. Figure 7 shows a typical design flow with the ispLEVER design tool.

Figure 7 – ispGDX2 Design Tools Flow



Lattice Semiconductor Corporation, L (& design), Lattice (& design), in-system programmable, ispLEVER, ispGDX2, E2CMOS, ISP, sysCLOCK, sysIO, sysHSI, Performance Analyst, ispVM, ispUPDATE and specific product designations are either registered trademarks or trademarks of Lattice Semiconductor Corporation or its subsidiaries in the United States and/or other countries.

GENERAL NOTICE: Other product names used in this publication are for identification purposes only and may be trademarks of their respective holders.