ispMACH 5000VG Timing Model Design and Usage Guidelines

November 2001 Technical Note TN1001

Introduction

Understanding how the placement of the design influences timing is essential when designing into the ispMACH™ 5000VG family. A signal in the device can take several paths, where each different path affects timing in some manner. This application note explains the ispMACH 5000VG timing model and offers a few techniques to enhance speed in the design when using this timing model.

ispMACH 5000VG Architecture Basics

The ispMACH 5000VG family architecture consists of multiple SuperWIDE™ 68-input, 32-macrocell Generic Logic Blocks (GLBs) interconnected through a tiered routing system. Groups of four GLBs, referred to as segments, are interconnected with a Segment Routing Pool (SRP). Segments are interconnected via the Global Routing Pool (GRP). In addition to these components, the ispMACH 5000VG has I/O cells and Phase Locked Loops (PLLs). Together the GLBs and the routing pools allow designers to create large designs in a single device without compromising performance. See Figures 1 and 2 for the ispMACH 5000VG architecture diagram and the segment diagram, respectively.

Figure 1. ispMACH 5000VG Architecture Diagram

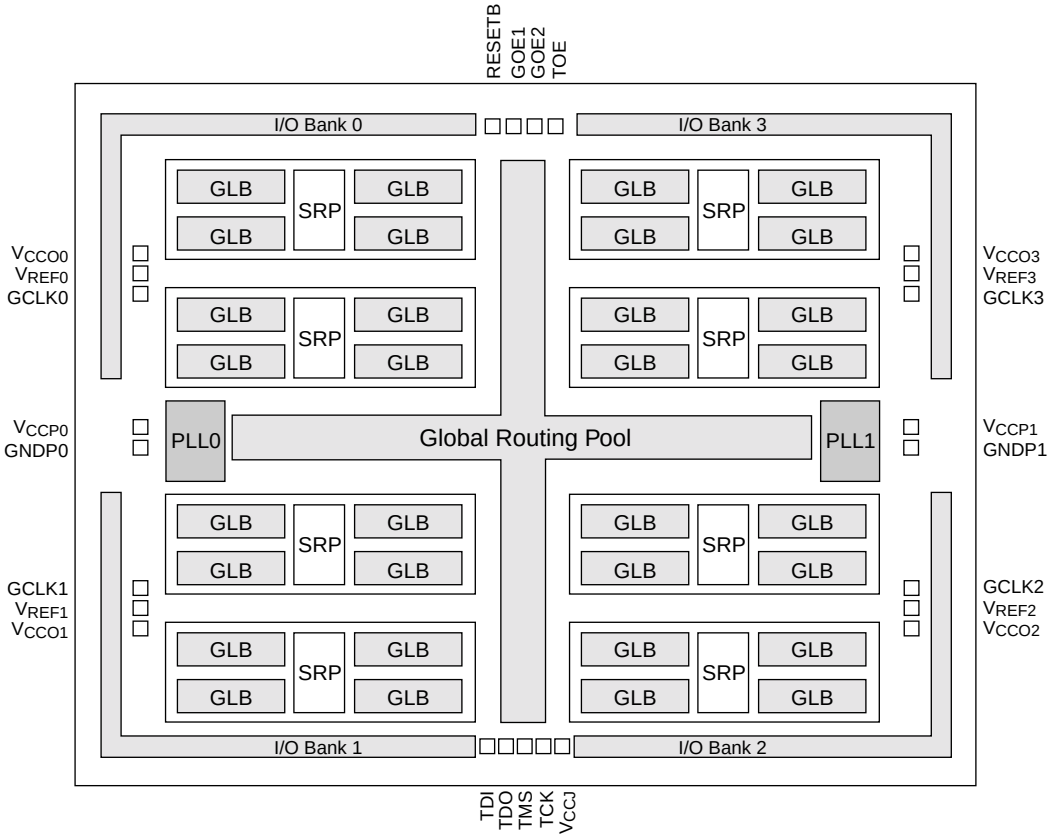
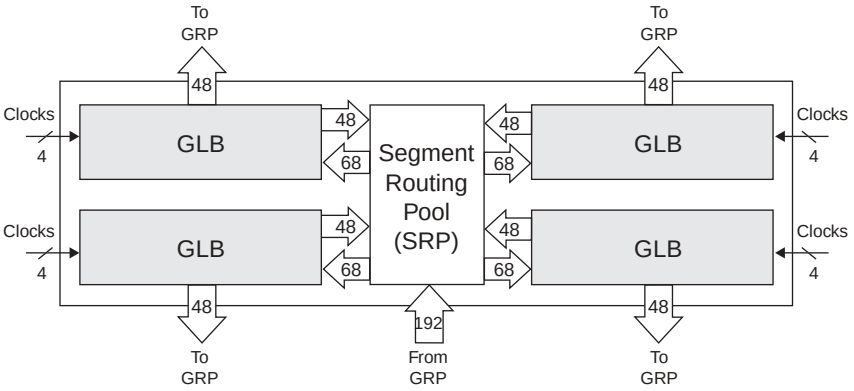


Figure 2. ispMACH 5000VG Segment Diagram



Each GLB contains 32 macrocells, a Clock Generator, a programmable AND-array with 160 logic product terms and three control product terms, a Dual-OR Array (DOA), and a Product Term Sharing Array (PTSA). The GLB has 68 inputs coming from the routing scheme, which is the SRP. The Clock Generator in the ispMACH 5000VG family provides clock selection within the GLB. These inputs are available in both true and complement form for every product term. The three control product terms are used for shared reset, clock and output enable functions. Moreover, the GLB contains 163 product terms. These product terms form groups of five product term clusters, which feed the PT sharing array or the macrocell directly. The ispMACH 5000VG allows up to 160 product terms to be connected to a single macrocell via the product term expanders and PT Sharing Array.

The macrocell, which is in the GLB, is designed to provide flexible clocking and control functionality with the capability to select between global, product term, and block level resources. The outputs of the macrocells are feedback into the switch matrices and, if required, the sysIO™ cell.

The sysIO Cell contains an output enable (OE) generator, a programmable tri-state output buffer, a programmable input buffer, a programmable pull-up resistor, a programmable pull-down resistor, and a programmable bus-friendly latch. The sysIO Cell receives its input from the 16 outputs of the GLB. The OE Generator selects between the PTOE, Shared PTOE, GOE0 and GOE1 for increased flexibility. The four Shared PTOE signals are derived from PT163 of each GLB in the segment. The PTOE signal is derived from the first product term in each macrocell cluster, which is directly routed to the OE multiplexer. Therefore, every sysIO Cell can have a different OE signal. The output of the OE multiplexer goes through a logical AND with the TOE signal to allow easy tri-stating of the outputs for testing purposes. The output of the I/O cell feeds back to its associated macrocell and a direct path to the GRP and SRP.

All I/Os in the ispMACH 5000VG family are sysIO, which are split into four banks and can be configured to different I/O standards, drive strengths and slew rates. Each bank has a separate I/O power supply and reference voltage. Inputs can be set to a variety of standards providing the reference voltage requirements of the chosen standards are compatible. The enhanced output enable multiplexor provides up to 14 different output enable choices per sysIO cell. Within a bank, the outputs can be set to differing standards providing the I/O power supply voltage and the reference voltage requirements of the chosen standard are compatible. Support for this variety of standards allows designers to achieve significantly higher board level performance compared to the more traditional LVCMOS standards. The input and output buffers utilize the sysIO feature to optimize performance and reduce complexity of external routing.

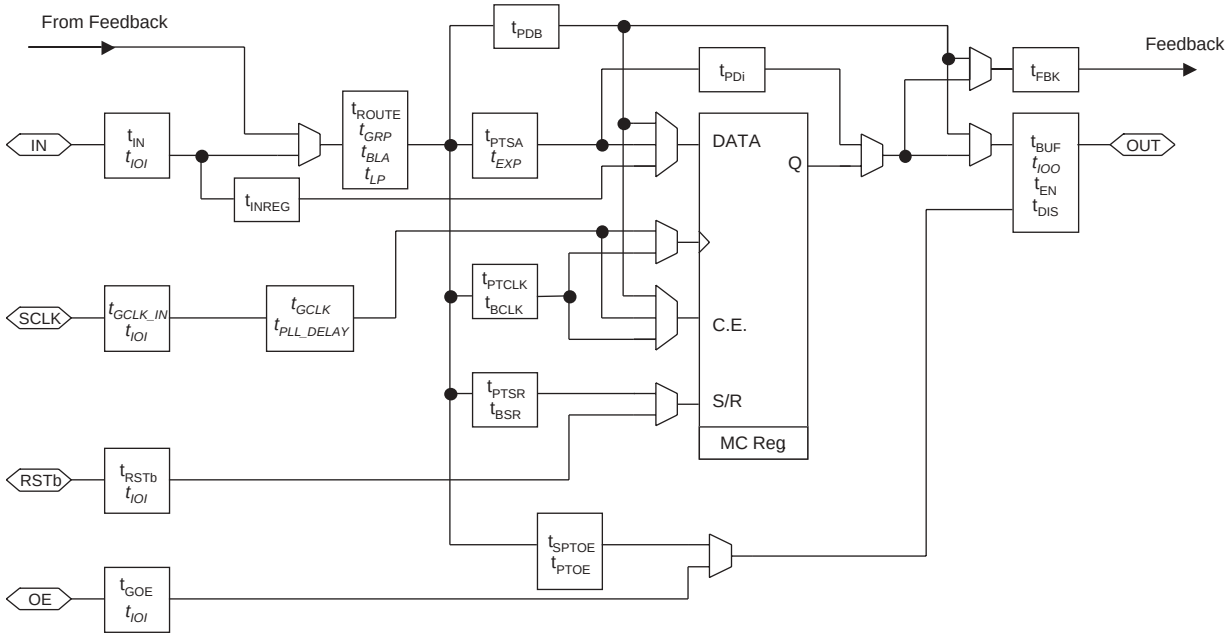
The ispMACH5000VG devices also contain sysCLOCK™ PLLs that provide designers with increased clocking flexibility. The PLLs can be used to synthesize new clocks for use on-chip or elsewhere within the system. Plus, they can be used to de-skew clocks, again both at the chip and system level. A variable delay line capability further improves this and allows designers to retard or advance the clock to tune set-up and clock-to-out times for optimal results. The operation of the PLL block synchronizes the input and feedback signals, performs clock synthesis, and provides duty cycle correction. Furthermore, the PLL has a programmable delay feature, t_{PLL_DELAY}, to allow for additional timing control.

For additional information on the architecture, please refer to the ispMACH 5000VG data sheet.

ispMACH 5000VG Timing Model

The main purpose of the ispMACH 5000VG timing model is to accurately depict the various timing paths and delays for the device. As shown in Figure 3, each logic element has its own set of delay parameters. Each delay element represents a subset of the architecture.

Figure 3. ispMACH 5000VG Timing Model



Note: Italicized parameters are delay adders that may be required, dependent on device configuration.

Timing Parameters

The timing model consists of internal parameters that describe the timing of architectural elements in the device. The software combines these internal parameters to calculate the external timing parameters. However, internal timing parameters are not tested or guaranteed. Only the external timing parameters are tested and guaranteed.

Internal Parameters

Table 1 lists the internal timing parameters for the ispMACH 5000VG. The table is split into several sections. The largest section is the “Routing Delays” section, which defines all of the timing delays that are a result of a signal propagating through a particular architectural feature. For example, t_{IN} represents the time it takes for a signal to propagate from the device I/O pad through the input buffer. The parameter t_{IN} is shown in the timing model in the same block as the parameter t_{IOI}, where t_{IOI} is an optional parameter. The parameters in italics are optional parameters in the model. These optional parameters are given for those features, such as programmable I/O interface standards, that may affect timing but are not always used or required.

Table 1. ispMACH 5000VG Family Timing Parameters

Parameter	Description
In/Out Delays	
t_{IN}	Input Buffer Delay
t_{GCLK_IN}	Global Clock Input Buffer Delay
t_{GOE}	Global OE Pin Delay
t_{BUF}	Delay through Output Buffer
t_{EN}	Output Enable Time
t_{DIS}	Output Disable Time
t_{RSTb}	Global RESETbar Pin Delay
Routing Delays	
t_{ROUTE}	Delay through SRP
t_{PTSA}	Product Term Sharing Array Delay
t_{PDB}	5-PT Bypass Propagation Delay
t_{PDi}	Macrocell Propagation Delay
t_{INREG}	Input Buffer to Macrocell Register Delay
t_{FBK}	Internal Feedback Delay
t_{GCLK}	Global Clock Tree Delay
t_{PLL_DELAY}	Programmable PLL Delay
t_{GRP}	Global Routing Pool Delay
t_{SPTOE}	Segment PT OE Delay
t_{PTOE}	Macrocell PT OE Delay
Register/Latch Delays	
t_{S}	D-Register Setup Time
t_{H}	D-Register Hold Time
t_{COi}	Register Clock to Output/Feedback MUX Time
t_{CES}	Clock Enable Setup Time
t_{CEH}	Clock Enable Hold Time
t_{SL}	Latch Setup Time
t_{HL}	Latch Hold Time
t_{GOi}	Latch Gate to Output/Feedback MUX Time
t_{PDLi}	Propagation Delay through Transparent Latch to Output/Feedback MUX
t_{SRi}	Asynchronous Reset or Set to Output/Feedback MUX Delay
t_{SRR}	Asynchronous Reset or Set Recovery Delay
Control Delays	
t_{BCLK}	GLB PT Clock Delay
t_{PTCLK}	Macrocell PT Clock Delay
t_{BSR}	Block PT Set/Reset Delay
t_{PTSR}	Macrocell PT Set/Reset Delay

Table 2. ispMACH 5000VG Family Timing Adders

Adder Type	Base Parameter	Description
t_{BLA}	t_{ROUTE}	GLB Loading Adder
t_{EXP}	t_{PTSA}	PT Expander Adder
t_{LP}	t_{ROUTE}	Low Power Adder
t_{IOI} Input Adders		
LVCMOS18_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using LVCMOS1.8 standard
LVCMOS25_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using LVCMOS2.5 standard
LVCMOS33_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using LVCMOS3.3standard
PCI_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using PCI standard
PCI_X_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using PCI_X standard
AGP_1X_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using AGP-1X standard
SSTL3_I_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using SSTL3_I standard
SSTL3_II_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using SSTL3_II standard
SSTL2_I_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using SSTL2_I standard
SSTL2_II_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using SSTL2_II standard
CTT33_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using CTT3.3 standard
CTT25_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using CTT2.5 standard
HSTL_I_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using HSTL_I standard
HSTL_II_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using HSTL_II standard
GTL+_in	t_{IN}, t_{GCLK_IN}, t_{RSTb}, t_{GOE}	Using GTL+ standard
LVDS_in	t_{IN}, t_{GCLK_IN}	Using LVDS standard
LVPECL	t_{IN}, t_{GCLK_IN}	Using LVPECL standard
t_{IOO} Output Adders		
LVCMOS18_4mA_out	t_{BUF}, t_{EN}	Output configured as 1.8V & 4mA Buffer
LVCMOS18_5.33mA_out	t_{BUF}, t_{EN}	Output configured as 1.8V & 5.33mA Buffer
LVCMOS18_8mA_out	t_{BUF}, t_{EN}	Output configured as 1.8V & 8mA Buffer
LVCMOS18_12mA_out	t_{BUF}, t_{EN}	Output configured as 1.8V & 12mA Buffer
LVCMOS25_4mA_out	t_{BUF}, t_{EN}	Output configured as 2.5V & 4mA Buffer
LVCMOS25_5.33mA_out	t_{BUF}, t_{EN}	Output configured as 2.5V & 5.33mA Buffer
LVCMOS25_8mA_out	t_{BUF}, t_{EN}	Output configured as 2.5V & 8mA Buffer
LVCMOS25_12mA_out	t_{BUF}, t_{EN}	Output configured as 2.5V & 12mA Buffer
LVCMOS25_16mA_out	t_{BUF}, t_{EN}	Output configured as 2.5V & 16mA Buffer
LVCMOS33_4mA_out	t_{BUF}, t_{EN}	Output configured as 3.3V & 4mA Buffer
LVCMOS33_5.33mA_out	t_{BUF}, t_{EN}	Output configured as 3.3V & 5.33mA Buffer
LVCMOS33_8mA_out	t_{BUF}, t_{EN}	Output configured as 3.3V & 8mA Buffer
LVCMOS33_12mA_out	t_{BUF}, t_{EN}	Output configured as 3.3V & 12mA Buffer
LVCMOS33_16mA_out	t_{BUF}, t_{EN}	Output configured as 3.3V & 16mA Buffer
LVCMOS33_20mA_out	t_{BUF}, t_{EN}	Output configured as 3.3V & 20mA Buffer
Slow Slew	t_{BUF}, t_{EN}	Output configured for slow slew rate
PCI_out	t_{BUF}, t_{EN}	Using PCI standard
PCI_X_out	t_{BUF}, t_{EN}	Using PCI-X standard
AGP_1X_out	t_{BUF}, t_{EN}	Using AGP-1X standard
SSTL3_I_out	t_{BUF}, t_{EN}	Using SSTL3_I standard
SSTL3_II_out	t_{BUF}, t_{EN}	Using SSTL3_II standard
SSTL2_I_out	t_{BUF}, t_{EN}	Using SSTL2_I standard

Table 2. ispMACH 5000VG Family Timing Adders (Continued)

Adder Type	Base Parameter	Description
SSTL2_II_out	t_{BUF}, t_{EN}	Using SSTL2_II standard
CTT33_out	t_{BUF}, t_{EN}	Using CCT3.3 standard
CTT25_out	t_{BUF}, t_{EN}	Using CCT2.5 standard
HSTL_I_out	t_{BUF}, t_{EN}	Using HSTL_I standard
HSTL_II_out	t_{BUF}, t_{EN}	Using HSTL_II standard
GTL+_out	t_{BUF}, t_{EN}	Using GTL+ standard

External Parameters

When calculating timing for the ispMACH 5000VG, timing delays are associated with each block in the architecture. Register setup and hold times are calculated using the path delays on the data and clock signals into the register in conjunction with the inherent setup and hold times of the register itself. By using the internal path delays, accurate times are derived for timing with respect to the device input and output pins.

Below are some basic derivations of the more common external timing parameters. Note that the software will report the setup and hold times as being 0ns when the values are calculated as negative.

Setup Times

Setup time = Logic Delay + t_S - Clock Delay

$t_{S\_BYPASS} = (t_{IN} + t_{ROUTE} + t_{PDB}) + t_S - (t_{GCLK\_IN} + t_{GCLK})$ Synchronous Setup
$t_{S\_PTSA} = (t_{IN} + t_{ROUTE} + t_{PTSA}) + t_S - (t_{GCLK\_IN} + t_{GCLK})$

$t_{SA} = (t_{IN} + t_{ROUTE} + t_{PTSA}) + t_S - (t_{IN} + t_{ROUTE} + t_{PTCLK})$ Asynchronous Setup
$t_{SIR} = (t_{IN} + t_{INREG}) + t_S - (t_{GCLK\_IN} + t_{GCLK})$ Input Register Setup

Hold Times

Hold Time = Clock Delay + t_H - Logic Delay

$t_{H\_BYPASS} = (t_{GCLK\_IN} + t_{GCLK}) + t_H - (t_{IN} + t_{ROUTE} + t_{PDB})$ Synchronous Hold
$t_{H\_PTSA} = (t_{GCLK\_IN} + t_{GCLK}) + t_H - (t_{IN} + t_{ROUTE} + t_{PTSA})$

$t_{HA} = (t_{IN} + t_{ROUTE} + t_{PTCLK}) + t_H - (t_{IN} + t_{ROUTE} + t_{PTSA})$ Asynchronous Hold
$t_{HIR} = (t_{GCLK\_IN} + t_{GCLK}) + t_H - (t_{IN} + t_{INREG})$ Input Register Hold

Clock-to-Out Time

Clock-to-Out Time = Clock Delay + t_{COi} + Output Path Delay
$t_{COS} = t_{GCLK\_IN} + t_{GCLK} + t_{COi} + t_{BUF}$

Combinatorial Propagation Times

$t_{PD} = t_{IN} + t_{ROUTE} + t_{PDB} + t_{BUF}$
$t_{PD\_PTSA} = t_{IN} + t_{ROUTE} + t_{PTSA} + t_{PDi} + t_{BUF}$
$t_{PD\_GLOBAL} = t_{PD\_PTSA} + t_{GRP}$
$= t_{IN} + t_{ROUTE} + t_{PTSA} + t_{PDi} + t_{BUF} + t_{GRP}$

Reset Time

$t_R = t_{RSTB} + t_{SRi} + t_{BUF}$

Input-to-Output Times

$t_{LPTOE/DIS} = t_{IN} + t_{ROUTE} + t_{PTOE} + t_{EN}$
$t_{SPTOE/DIS} = t_{IN} + t_{ROUTE} + t_{SPTOE} + t_{EN}$
$t_{GOE/DIS} = t_{GOE} + t_{EN}$

Maximum Frequency

$f_{MAX}(Ext.) = 1/(t_{CO} + t_{S\_PTSA})$
$f_{MAX}(Int.) = 1/(t_{COi} + t_{FBK} + t_{ROUTE} + t_S)$

Examples

The three examples beneath demonstrate using the ispMACH 5000VG timing model. The first example is a combinatorial logic design and it shows the use of internal feedback. The second example, a synchronous sequential logic design, illustrates how to calculate f_{MAX}. Lastly, the third example explains a timing path using t_{GRP}

Example 1

This combinatorial logic design is fit into an ispMACH5000VG. A group of input signals are routed to Block A, which is in high power mode. Logic is generated in array “A” and allocated to Macrocell A5, which is configured as a combinatorial path. This logic is sent to pad I/O 6, which is configured for a slow slew rate. The signal delay T1 of this path would be:

$$T_1 = t_{IN} + t_{ROUTE} + t_{PTSA} + t_{PDi} + t_{BUF} + t_{IOO(SLEW)}$$

This logic is also fed back to the GRP via the internal feedback path and then routed to Block D. A second logic is generated in array “D” using the first logic along with another group of input signals. This second logic is allocated to Macrocell D8, which is configured as a combinatorial path. This second logic is sent to pad I/O 31, which is in fast slew rate. The longest delay path of this design would be from Block A to I/O 31 and the delay $T_{CRITICAL}$ is:

$$T_{CRITICAL} = t_{IN} + t_{ROUTE} + t_{PTSA} + t_{PDi} + t_{ROUTE} + t_{PTSA} + t_{PDi} + t_{BUF}$$

When the number of product terms is increased beyond 20, the timing will change. T_{EXP} is used when more than 20 product terms are needed and the software setting. The longest path would be for an input signal to utilize the expander feature of the ispMACH 5000VG.

$$T_2 = t_{IN} + t_{ROUTE} + t_{PTSA} + t_{EXP} + t_{PDi} + t_{BUF} + t_{IOO(SLEW)}$$

Example 2

This synchronous sequential logic design has a 16-bit up counter with load enable and reset. It fits into an ispMACH 5000VG using 16 macrocells configured with T-type registers. Register inputs are defined by the device inputs and flip-flop output, which is internally fed back to the GRP. Under these conditions, the period t_{CNT} is limited by the internal delay from the flip-flop outputs through the internal feedback and logic to the flip-flop inputs:

$$t_{CNT} = t_{COi} + t_{FBK} + t_{ROUTE} + t_{PTSA} + t_S$$

And the f_{MAX} is designated “f_{MAXINT}”:

$$f_{MAXINT} = 1/ t_{CNT}$$

Example 3

This combinatorial logic design using an ispMACH 5000VG is a simple segment-to-segment calculation. This signal t_{PATH} starts from GLB A of Segment 1, passes through the SRP of Segment 1, enters into the SRP of Segment 2 and ends in GLB B of Segment 2. The following is a possible calculation:

$$t_{PATH} = t_{IN} + t_{ROUTE} + t_{GRP} + t_{PDB} + t_{BUF}$$

Designing for Speed

While fitting a design into an ispMACH 5000VG device, the user needs to consider the fastest possible speed and shortest path within the device. Utilize the following hints and tips to obtain faster speed:

- Specify the quickest register logic to get a higher f_{MAX}.
- Minimize logic between registers to get the shortest register-to-register path.
- Utilize the 5-PT cluster bypass when possible.
- Minimize the number of routes going from one segment to another.

The path taken between the registers in the ispMACH 5000VG depends on the number of product terms. For 5 PTs, the signal will go through the 5-PT cluster bypass. For 6 to 35 PTs, the product terms will be allocated to the macrocells using the PTSA. If the number of product terms exceeds 35, the user will need to utilize the PT expander in the ispMACH 5000VG. This device allows up to 160 product terms to be connected to a single expander.

Conclusion

The ispMACH 5000VG timing model gives a clearer understanding of the timing calculation. This timing model illustrates both the internal and external feedback paths. When using the timing model, the user can control the critical path timing in a high-speed design.

Technical Support Assistance

Hotline: 1-800-LATTICE (Domestic)
1-408-826-6002 (International)
e-mail: techsupport@latticesemi.com