



XAPP456 (v1.0) March 13, 2007

Custom PCI Timing Budgets for Spartan-3 Generation FPGAs

Author: Eric Crabill

Summary

The PCI Local Bus Specification, Revision 3.0 ("the PCI specification"), defines two timing budgets. One timing budget is for use with 33 MHz operation, and the other timing budget is for use with 66 MHz operation. These two timing budgets define the I/O timing parameters for compliant 33 MHz and 66 MHz components.

In open systems, compliance with the PCI specification is a requirement to ensure interoperability. However, in embedded designs, it is possible to create custom timing budgets that enable system designers to do one or more of the following:

- Reduce total system cost by using less expensive devices
- Achieve higher data transfer rates than allowed by specification
- Add more loads to the bus to accommodate additional devices and connectors
- Increase the physical length of the bus to accommodate novel bus topologies

The information presented in this application note is applicable to any embedded PCI implementation using Xilinx FPGA devices. The provided example applies this information to a design using Xilinx Spartan™-3 Generation FPGAs with Xilinx LogiCORE™ PCI interfaces.

Note: Xilinx LogiCORE PCI interfaces are designed for compliance to the PCI Local Bus Specification when implemented in the devices and at the bus frequencies listed in the relevant product data sheets. Deviations from the implementations listed in the data sheets are not supported. Designers must take all necessary steps and precautions to verify that modifications work as intended in their designs.

PCI Timing Budgets

Table 1 shows the 33 MHz and 66 MHz timing budgets from the PCI specification. This table includes some computed parameters for signal propagation delays.

Table 1: Specification Timing Budgets

Timing Parameter	33 MHz	66 MHz
T_{CYC} min	30 ns	15 ns
T_{VAL} max	11 ns	6 ns
$T_{VAL-PTP}$ max	12 ns ⁽¹⁾	6 ns ⁽¹⁾
T_{VAL} min	2 ns	2 ns ⁽²⁾
$T_{VAL-PTP}$ min	2 ns ⁽¹⁾	2 ns ^(1,2)
T_{ON} min	2 ns	2 ns ⁽²⁾
T_{OFF} max	28 ns	14 ns
T_{PROP} max	10 ns ⁽³⁾	5 ns ⁽³⁾
$T_{PROP-PTP-REQ}$ max	4 ns ⁽³⁾	3 ns ⁽³⁾

© 2007 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

LIMITATION OF LIABILITY: In no event will Xilinx or its licensors be liable for any loss of data, lost profits, cost or procurement of substitute goods or services, or for any special, incidental, consequential, or indirect damages arising from the use or operation of the guidelines or accompanying documentation, however caused and on any theory of liability. This limitation will apply even if Xilinx has been advised of the possibility of such damage. This limitation shall apply notwithstanding the failure of the essential purpose of any limited remedies herein.

Table 1: Specification Timing Budgets (Continued)

Timing Parameter	33 MHz	66 MHz
$T_{PROP-PTP-GNT} \text{ max}$	6 ns ⁽³⁾	3 ns ⁽³⁾
$T_{PROP} \text{ min}$	0 ns ⁽⁴⁾	0 ns ⁽⁴⁾
$T_{PROP-PTP} \text{ min}$	0 ns ⁽⁴⁾	0 ns ⁽⁴⁾
$T_{SU} \text{ min}$	7 ns	3 ns
$T_{SU-PTP-GNT} \text{ min}$	10 ns	5 ns
$T_{SU-PTP-REQ} \text{ min}$	12 ns	5 ns
$T_H \text{ min}$	0 ns	0 ns
$T_{H-PTP} \text{ min}$	0 ns	0 ns
$T_{SKEW} \text{ max}$	2 ns	1 ns

Notes:

1. REQ# and GNT# are point-to-point signals and can use relaxed output drivers.
2. Can be reduced to 1 ns while M66EN is asserted due to lower $T_{SKEW} \text{ max}$.
3. Computed or not explicitly listed in the PCI specification.
4. A conservative assumption based on physical impossibility.

PCI is a synchronous bus. When two or more devices are bussed together with a common clock, new synchronous paths are created. These synchronous paths are created between the output flip-flops in one device and the input flip-flops in all other devices. These flip-flops are all clocked by the common bus clock. The PCI specification permits a certain amount of non-constructive clock skew ($T_{SKEW} \text{ max}$) between devices.

Bussed Signals versus Point-to-Point Signals

All PCI signals, except those used for arbitration, are considered bussed signals. Bussed signals are connected to the corresponding pins on every device. Even if a PCI implementation has only two devices, the signals are still called bussed signals!

The PCI signals used for arbitration, REQ# and GNT#, are considered point-to-point signals. In a PCI implementation with N devices, there usually exists an arbiter with N pairs of REQ# and GNT# pins, one pair for each device. As a result, these signals are never loaded with more than one driver and one receiver, and they are always actively driven during normal operation. The drivers have relaxed electrical requirements, and the timing requirements are also relaxed in some cases.

Timing Budget for Bussed Signals

In the span of one cycle ($T_{CYC} \text{ min}$), information must travel from the output flip-flops of one device ($T_{VAL} \text{ max}$), across the bus ($T_{PROP} \text{ max}$), and be set up ($T_{SU} \text{ min}$) at the input flip-flops in all other devices. Additionally, the information must not travel from the output flip-flops of one device ($T_{VAL} \text{ min}$) and across the bus ($T_{PROP} \text{ min}$) so quickly that it violates the hold requirement for the data from the previous cycle ($T_H \text{ min}$) at the input flip-flops in all the other devices.

In turnaround cycles where three-state drivers are transitioning to high impedance, the process must complete in advance of the cycle end ($T_{OFF} \text{ max}$). Conversely, when three-state drivers turn on, they must not turn on so quickly ($T_{ON} \text{ min}$) that they violate any hold requirement for the data from the previous cycle.

Equation 1 through Equation 4 define the key timing parameters:

$$T_{VAL}^{max} + T_{PROP}^{max} + T_{SU}^{min} + T_{SKEW}^{max} \leq T_{CYC}^{min} \quad \text{Equation 1}$$

$$T_{VAL}^{min} + T_{PROP}^{min} - T_{SKEW}^{max} \geq T_H^{min} \quad \text{Equation 2}$$

$$T_{OFF}^{max} + T_{SKEW}^{max} \leq T_{CYC}^{min} \quad \text{Equation 3}$$

$$T_{ON}^{min} + T_{PROP}^{min} - T_{SKEW}^{max} \geq T_H^{min} \quad \text{Equation 4}$$

The PCI specification indicates that only T_{PROP}^{max} is a computed parameter. Thus the other device parameters are “by definition”, and T_{PROP}^{max} can be obtained by subtracting the other parameters from the clock cycle time. A conservative value for T_{PROP}^{min} is zero.

For the computed value of T_{PROP}^{max} , the system designer is then free to design any bus topology that yields an actual propagation delay less than or equal to T_{PROP}^{max} . For this reason, the PCI specification does not call out a required topology or indicate a maximum number of “loads”. The system designer assumes all devices are compliant with the PCI specification and designs the bus topology to satisfy T_{PROP}^{max} . The PCI specification describes how to determine T_{PROP}^{max} based on simulated or measured data.

Timing Budget for Point-to-Point Signals

The timing budget for point-to-point signals is similar. It is not necessary to consider three-state transitions because these signals are always actively driven during normal operation. In the 66 MHz timing budget, the two point-to-point signals have the same timing requirements. However, in the 33 MHz timing budget, they have different input setup requirements with respect to the clock. Refer to $T_{SU-PTP-REQ}$ and $T_{SU-PTP-GNT}$ in Table 1.

Equation 5 and Equation 6 show the timing budget from device to arbiter (REQ#).

$$T_{VAL-PTP}^{max} + T_{PROP-PTP-REQ}^{max} + T_{SU-PTP-REQ}^{min} + T_{SKEW}^{max} \leq T_{CYC}^{min} \quad \text{Equation 5}$$

$$T_{VAL-PTP}^{min} + T_{PROP-PTP}^{min} - T_{SKEW}^{max} \geq T_{H-PTP}^{min} \quad \text{Equation 6}$$

Equation 7 and Equation 8 show the timing budget from arbiter to device (GNT#).

$$T_{VAL-PTP}^{max} + T_{PROP-PTP-GNT}^{max} + T_{SU-PTP-GNT}^{min} + T_{SKEW}^{max} \leq T_{CYC}^{min} \quad \text{Equation 7}$$

$$T_{VAL-PTP}^{min} + T_{PROP-PTP}^{min} - T_{SKEW}^{max} \geq T_{H-PTP}^{min} \quad \text{Equation 8}$$

As before, only $T_{PROP-PTP-REQ}^{max}$ and $T_{PROP-PTP-GNT}^{max}$ are computed parameters, and a conservative value for $T_{PROP-PTP}^{min}$ is zero. For the given values of $T_{PROP-PTP-REQ}^{max}$ and $T_{PROP-PTP-GNT}^{max}$, the system designer is then free to design any bus topology that yields actual propagation delays less than or equal to $T_{PROP-PTP-REQ}^{max}$ and $T_{PROP-PTP-GNT}^{max}$.

Custom Timing Budgets

The PCI specification suggests that system designers use compliant components and increase the cycle time, T_{CYC}^{min} , in order to increase T_{PROP}^{max} . An increase in T_{PROP}^{max} allows system designers to add more loads to the bus or to increase the physical length of the bus.

Although not discussed in the PCI specification, it is possible to use devices with I/O timing that differs from the PCI specification requirements. Most ASSP vendors specify their I/O performance to comply with the 33 MHz or 66 MHz timing specifications. However, with an FPGA-based design, it is possible to achieve a wide range of I/O performance. Using this information with a custom timing budget allows many possibilities. The only requirement to ensure proper synchronous operation of the bus is that Equation 9 through Equation 16 are satisfied. These equations are based on Equation 1 through Equation 8.

$$T_{VALmax,tx} + T_{PROPmax} + T_{SUmin,rx} + T_{SKEWmax} \leq T_{CYCmin} \quad \text{Equation 9}$$

$$T_{VALmin,tx} + T_{PROPmin} - T_{SKEWmax} \geq T_{Hmin,rx} \quad \text{Equation 10}$$

$$T_{OFFmax,tx} + T_{SKEWmax} \leq T_{CYCmin} \quad \text{Equation 11}$$

$$T_{ONmin,tx} + T_{PROPmin} - T_{SKEWmax} \geq T_{Hmin,rx} \quad \text{Equation 12}$$

$$T_{VAL-PTPmax} + T_{PROP-PTP-REQmax} + T_{SU-PTP-REQmin} + T_{SKEWmax} \leq T_{CYCmin} \quad \text{Equation 13}$$

$$T_{VAL-PTPmin} + T_{PROP-PTPmin} - T_{SKEWmax} \geq T_{H-PTPmin} \quad \text{Equation 14}$$

$$T_{VAL-PTPmax} + T_{PROP-PTP-GNTmax} + T_{SU-PTP-GNTmin} + T_{SKEWmax} \leq T_{CYCmin} \quad \text{Equation 15}$$

$$T_{VAL-PTPmin} + T_{PROP-PTPmin} - T_{SKEWmax} \geq T_{H-PTPmin} \quad \text{Equation 16}$$

For every possible driver (tx) on the bus, the relationships in Equation 9 through Equation 16 must be checked with respect to every possible receiver (rx) on the bus. In the general case, pin-to-pin timing parameters may be different for each device. The parameter names in these equations explicitly indicate which device (driver or receiver) is contributing the term to the equation. For example, a bus with five devices, where each device has unique parameters, may require the evaluation of up to 40 equations.

Example: Reduce Total System Cost

Consider an embedded system design that consists of an ASSP host bridge and an FPGA peripheral device. The devices communicate using a 32-bit, 66 MHz bus. The host bridge has an integrated arbiter, and the bus always runs at 66 MHz (that is, M66EN is always asserted). The original design used the XC3S500E-FT256-5 device because this device was listed in the Xilinx LogiCORE PCI32 Interface Data Sheet as supporting a 32-bit, 66 MHz interface. The other logic in the device could have been implemented with a slower, less expensive device. Is it possible to redesign with the XC3S500E-FT256-4 device to reduce cost?

There are two ways to approach this problem. One approach is to extract as much performance out of the I/O timing as possible and then compute the allowed T_{PROP} . This method provides the best margin for board layout but the FPGA implementation effort may be considerable. Another approach is to extract as much performance out of T_{PROP} as possible and then compute the allowed I/O timing. This method provides the best margin for the FPGA design but the PCB layout effort may be considerable.

This example uses the first approach. Starting with the design files for a 32-bit, 33 MHz interface for the XC3S500E-FT256-4 device, the SelectIO mode is changed from PCI33_3 to PCI66_3 to maintain electrical compliance. The original constraint file for 33 MHz operation with the Xilinx LogiCORE PCI32 interface specifies the following timing requirements:

```
#####
# Time Specs (Compliant 33 MHz)
#####
#
# Important Note: The timespecs used in this section cover all possible
# paths. Depending on the design options, some of the timespecs may
# not contain any paths. Such timespecs are ignored by PAR and TRCE.
#
#           1) Clock to Output      =      11.000 ns
#           2) Setup                 =       7.000 ns
#           3) Grant Setup           =      10.000 ns
#           4) Datapath Tristate     =      28.000 ns
#           5) Period                =      30.000 ns
#
# Note: Timespecs are derived from the PCI Bus Specification. Use of
# offset constraints allows the timing tools to automatically include
# the clock delay estimates. These constraints are for 33 MHz operation.
```

```

#
# The following timespecs are for setup.
#
TIMEGRP "PCI_PADS_D" OFFSET = IN 7.000 VALID 7.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = IN 7.000 VALID 7.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = IN 7.000 VALID 7.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_C" OFFSET = IN 7.000 VALID 7.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_G" OFFSET = IN 10.000 VALID 10.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
#
# The following timespecs are for clock to out where stepping is not used.
#
TIMEGRP "PCI_PADS_D" OFFSET = OUT 11.000 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = OUT 11.000 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = OUT 11.000 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_C" OFFSET = OUT 11.000 AFTER "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_G" OFFSET = OUT 11.000 AFTER "PCLK" TIMEGRP "ALL_FFS" ;
#
# The following timespecs are for clock to out where stepping is used.
#
TIMEGRP "PCI_PADS_D" OFFSET = OUT 28.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = OUT 28.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = OUT 28.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
#
# The following timespec is for the period.
#
NET "PCLK" PERIOD = 30.000 ;
#
#####

```

The goal is to tighten the constraints as much as possible. The performance required for the compliant 66 MHz timing budget cannot be achieved in the slower XC3S500E-FT256-4 device, but it is possible to get close, in excess of the performance required for a compliant 33 MHz timing budget. For reference, a constraint file for 66 MHz operation with the Xilinx LogiCORE PCI32 interface specifies the following timing requirements:

```

#####
# Time Specs (Compliant 66 MHz)
#####
#
# Important Note: The timespecs used in this section cover all possible
# paths. Depending on the design options, some of the timespecs may
# not contain any paths. Such timespecs are ignored by PAR and TRCE.
#
#           1) Clock to Output      =      6.000 ns
#           2) Setup                  =      3.000 ns
#           3) Grant Setup            =      5.000 ns
#           4) Datapath Tristate     =     14.000 ns
#           5) Period                 =     15.000 ns
#
# Note: Timespecs are derived from the PCI Bus Specification. Use of
# offset constraints allows the timing tools to automatically include
# the clock delay estimates. These constraints are for 66 MHz operation.
#
# The following timespecs are for setup.
#
TIMEGRP "PCI_PADS_D" OFFSET = IN 3.000 VALID 3.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = IN 3.000 VALID 3.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = IN 3.000 VALID 3.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_C" OFFSET = IN 3.000 VALID 3.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_G" OFFSET = IN 5.000 VALID 5.000 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
#
# The following timespecs are for clock to out where stepping is not used.
#
TIMEGRP "PCI_PADS_D" OFFSET = OUT 6.000 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = OUT 6.000 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = OUT 6.000 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_C" OFFSET = OUT 6.000 AFTER "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_G" OFFSET = OUT 6.000 AFTER "PCLK" TIMEGRP "ALL_FFS" ;

```

```
#
# The following timespecs are for clock to out where stepping is used.
#
TIMEGRP "PCI_PADS_D" OFFSET = OUT 14.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = OUT 14.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = OUT 14.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
#
# The following timespec is for the period.
#
NET "PCLK" PERIOD = 15.000 ;
#
#####
```

The original 33 MHz design easily meets the timing requirements. First, the PERIOD constraint is changed to 15 ns to evaluate if the internal logic can run with a 66 MHz clock. A pass through the implementation tools reveals ample slack for the clock period. At this point, an iterative search for improved I/O timing begins. For this exercise, a few items are worthwhile to note:

- The input setup and hold constraints, specified using the `OFFSET IN {X} VALID {Y}` syntax, always use the same value for X and Y. For more information on timing constraints, consult the Constraint Guide in the Xilinx ISE™ documentation.

In short, an input setup requirement is specified as X with a zero hold requirement. While it is possible to use devices with positive hold requirements, clock distribution skew and minimum delay parameters must be closely evaluated.

- The input signal paths usually can employ delay buffers. Typically, these are specified in the user constraint file. It may be advantageous to consider changes to the delay buffer settings. If you follow the recommendation to avoid positive hold requirements, the goal in delay buffer adjustment is to make the incurred delay as large as possible to guarantee a zero hold requirement, but no larger.
- It is not necessarily a good idea to maximize the performance from the I/O timing. Without the use of directed routing constraints or guide files, it may be difficult for the automatic router to consistently repeat the results if there is very little timing slack. It is wise to save a copy of a routed NCD that meets aggressive timing requirements so that the copy can be later used as a guide file or to generate directed routing constraints.

After several iterations of constraint changes, followed by implementation, the following constraints were found to consistently meet timing requirements and offer much higher performance:

```
#####
# Time Specs (Custom)
#####
#
# Important Note: The timespecs used in this section cover all possible
# paths. Depending on the design options, some of the timespecs may
# not contain any paths. Such timespecs are ignored by PAR and TRCE.
#
#           1) Clock to Output      =          6.450 ns
#           2) Setup                 =          3.800 ns
#           3) Grant Setup           =          3.800 ns
#           4) Datapath Tristate     =         14.000 ns
#           5) Period                =         15.000 ns
#
# Note: Use of offset constraints allows the timing tools to automatically
# include the clock delay estimates. This is a custom timing budget.
#
# The following timespecs are for setup.
#
TIMEGRP "PCI_PADS_D" OFFSET = IN  3.800 VALID 3.800 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = IN  3.800 VALID 3.800 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = IN  3.800 VALID 3.800 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_C" OFFSET = IN  3.800 VALID 3.800 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_G" OFFSET = IN  3.800 VALID 3.800 BEFORE "PCLK" TIMEGRP "ALL_FFS" ;
#
```

```
# The following timespecs are for clock to out where stepping is not used.
#
TIMEGRP "PCI_PADS_D" OFFSET = OUT 6.450 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = OUT 6.450 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = OUT 6.450 AFTER "PCLK" TIMEGRP "FAST_FFS" ;
TIMEGRP "PCI_PADS_C" OFFSET = OUT 6.450 AFTER "PCLK" TIMEGRP "ALL_FFS" ;
TIMEGRP "PCI_PADS_G" OFFSET = OUT 6.450 AFTER "PCLK" TIMEGRP "ALL_FFS" ;
#
# The following timespecs are for clock to out where stepping is used.
#
TIMEGRP "PCI_PADS_D" OFFSET = OUT 14.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
TIMEGRP "PCI_PADS_B" OFFSET = OUT 14.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
TIMEGRP "PCI_PADS_P" OFFSET = OUT 14.000 AFTER "PCLK" TIMEGRP "SLOW_FFS" ;
#
# The following timespec is for the period.
#
NET "PCLK" PERIOD = 15.000 ;
#
#####
```

As expected, the performance does not meet the PCI specification requirements for 66 MHz operation, but it is close. Table 2 lists the performance results of both devices on the bus in this design.

Table 2: Custom Timing Budget

Timing Parameter	XC3S500E FT256-4 Device	ASSP / Arbiter (66 MHz Compliant Device)
T _{CYC} min	15 ns	15 ns
T _{VAL} max	6.45 ns	6 ns
T _{VAL-PTP} max	6.45 ns	6 ns
T _{VAL} min	1 ns	1 ns
T _{VAL-PTP} min	1 ns	1 ns
T _{ON} min	1 ns	1 ns
T _{OFF} max	14 ns	14 ns
T _{PROP} max	TBD by calculation	TBD by calculation
T _{PROP-PTP-REQ} max	TBD by calculation	TBD by calculation
T _{PROP-PTP-GNT} max	TBD by calculation	TBD by calculation
T _{PROP} min	0 ns	0 ns
T _{PROP-PTP} min	0 ns	0 ns
T _{SU} min	3.8 ns	3 ns
T _{SU-PTP-GNT} min	3.8 ns	N/A (GNT# is an output)
T _{SU-PTP-REQ} min	N/A (REQ# is an output)	5 ns
T _H min	0 ns	0 ns
T _{H-PTP} min	0 ns	0 ns
T _{SKEW} max	1 ns	1 ns

Notes:

- XC3S500E-FT256-4 data for this illustrative example obtained using ISE 8.2i Service Pack 2 with Spartan-3E speed file version 1.25 and Xilinx LogiCORE PCI32 v3_160 interface. Total “exploration time” was 60 minutes. These values do not represent the absolute limit of performance, only what was easily obtainable in 60 minutes. Always use the most recent software and speed file version supported by the Xilinx LogiCORE PCI interfaces.

Both devices, in isolation, can run at a clock frequency of 66 MHz. However, the timing budget must be checked to ensure the synchronous system also operates at 66 MHz. For each device on the bus, an analysis must be performed from that device to each of the other devices on the bus. The evaluation yields the required T_{PROP} value for proper operation.

From the ASSP/Arbiter to the XC3S500E-FT256-4 device:

Equation 9: $6 \text{ ns} + T_{PROP \text{ max}} + 3.8 \text{ ns} + 1 \text{ ns} \leq 15 \text{ ns}$	TRUE	if $T_{PROP \text{ max}} \leq 4.2 \text{ ns}$
Equation 10: $1 \text{ ns} + 0 \text{ ns} - 1 \text{ ns} \geq 0 \text{ ns}$	TRUE	
Equation 11: $14 \text{ ns} + 1 \text{ ns} \leq 15 \text{ ns}$	TRUE	
Equation 12: $1 \text{ ns} + 0 \text{ ns} - 1 \text{ ns} \geq 0 \text{ ns}$	TRUE	
Equation 13: N/A		
Equation 14: N/A		
Equation 15: $6 \text{ ns} + T_{PROP-PTP-GNT \text{ max}} + 3.8 \text{ ns} + 1 \text{ ns} \leq 15 \text{ ns}$	TRUE	if $T_{PROP-PTP-GNT} \leq 4.2 \text{ ns}$
Equation 16: $1 \text{ ns} + 0 \text{ ns} - 1 \text{ ns} \geq 0 \text{ ns}$	TRUE	

From the XC3S500E-FT256-4 device to the ASSP/Arbiter:

Equation 9: $6.45 \text{ ns} + T_{PROP \text{ max}} + 3 \text{ ns} + 1 \text{ ns} \leq 15 \text{ ns}$	TRUE	if $T_{PROP \text{ max}} \leq 4.55 \text{ ns}$
Equation 10: $1 \text{ ns} + 0 \text{ ns} - 1 \text{ ns} \geq 0 \text{ ns}$	TRUE	
Equation 11: $14 \text{ ns} + 1 \text{ ns} \leq 15 \text{ ns}$	TRUE	
Equation 12: $1 \text{ ns} + 0 \text{ ns} - 1 \text{ ns} \geq 0 \text{ ns}$	TRUE	
Equation 13: $6.45 \text{ ns} + T_{PROP-PTP-REQ \text{ max}} + 5 \text{ ns} + 1 \text{ ns} \leq 15 \text{ ns}$	TRUE	if $T_{PROP-PTP-REQ} \leq 2.55 \text{ ns}$
Equation 14: $1 \text{ ns} + 0 \text{ ns} - 1 \text{ ns} \geq 0 \text{ ns}$	TRUE	
Equation 15: N/A		
Equation 16: N/A		

All the T_{PROP} results are positive, which is a good sign. Negative result values indicate that the proposed timing budget is clearly impossible. With positive result values, the next question to be answered is that of feasibility – can these T_{PROP} values be achieved?

The bussed signals must be constrained by a $T_{PROP \text{ max}}$ that is the worst case of all computed $T_{PROP \text{ max}}$ values, which is 4.2 ns. Compare this value to the $T_{PROP \text{ max}}$ value in the PCI specification timing budget, which is 5.0 ns. The physically short interconnect, the reduced loading, and the absence of connectors on the bus in this embedded design make the recovery of 800 ps from the $T_{PROP \text{ max}}$ portion of the timing budget feasible. To ensure success, this calculation must be verified prior to board fabrication using the appropriate simulation tools.

For the point-to-point GNT# signal, $T_{PROP-PTP-GNT \text{ max}}$ is also 4.2 ns. The PCI specification timing budget provides for a $T_{PROP-PTP-GNT \text{ max}}$ of 3.0 ns. There is more slack on GNT# with this custom timing budget than with the PCI specification timing budget.

For the point-to-point REQ# signal, $T_{PROP-PTP-REQ \text{ max}}$ is 2.55 ns. The PCI specification timing budget provides for a $T_{PROP-PTP-REQ \text{ max}}$ of 3.0 ns. There is a need to recover 450 ps from the $T_{PROP-PTP-REQ \text{ max}}$ portion of the timing budget. Again, the physically short interconnect and the absence of connectors on the bus in this embedded design make this feasible, but board-level timing simulation is critical to success.

A redesign is possible with the XC3S500E-FT256-4 device to reduce cost with no change to functionality.

Conclusion

System designers can take advantage of custom timing budgets to reduce cost, increase performance, and implement novel bus topologies. The example presented in this application note illustrates one of the possible applications. In practice, the timing budget can be manipulated in a variety of ways to achieve specific goals, as long as the underlying timing relationships are satisfied.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/13/07	1.0	Initial Xilinx release.