XILINX®

Using the XC9500 Timing Model

XAPP071 January, 1997 (Version 1.0)

Application Note

Summary

This application note describes how to use the XC9500 timing model.

Xilinx Family

XC9500

Introduction

All XC9500 CPLDs have a uniform architecture and an identical timing model, making them very easy to use and understand. To determine specific timing details, users need only compare their paths of interest to the architectural diagrams and, using the timing model presented here, perform a simple addition of incremental time delays.

Device Timing Overview

External signals arrive at the pins and are delivered through the I/O block to the FastCONNECT Switch Matrix. From the switch matrix, they are dispatched to the various Function Blocks (FBs). As the signals enter the FBs, they incur incremental time delays depending on how the signals are used within the FB. For example, all logic signals must pass through the AND array where they encounter product terms which add a time delay as the signals pass through. Additional time delay may be encountered if the signals pass through the cascade logic and are redirected toward macrocells that are further away than those directly attached to the product terms.

There are additional timing requirements such as setup and clock-to-output times involved with passing signals through a flip-flop. As the signals exit flip-flops, they either pass to the outside world, through the I/O pins, or are fed back into the FastCONNECT switch matrix for additional logic operations.

Design timing can be manually analyzed as separate signals, each having unique timing parameters that are easily calculated. However, the Xilinx software provides a detailed timing report that tallies and summarizes all paths specified by the designer. The timing report is based on the model described here and is a convenient text based mechanism for isolating and displaying timing relationships.

The timing model shown in Figure 1 is used by the M1 release of the Xilinx XACT*step* development software which provides complete fitters for the XC9500 family as well as the timing models for simulation and detailed static timing reports.

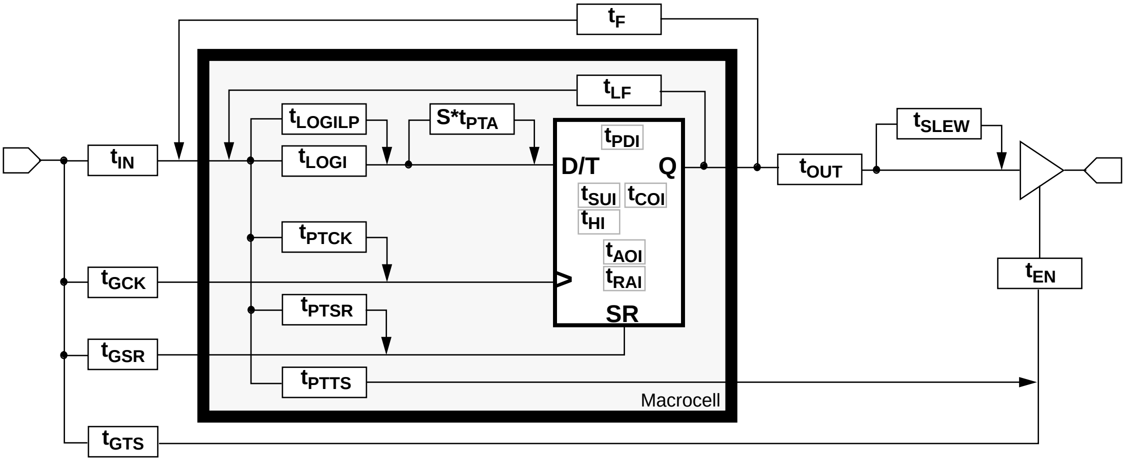


Figure 1: XC9500 Detailed Timing Model

Timing Model

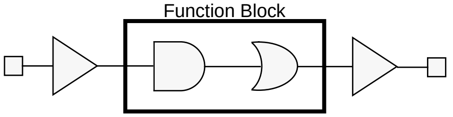
The timing model shown in Figure 1 resembles the XC9500 macrocell with additional time delays included to account for the FastCONNECT Switch Matrix and the I/O buffers. As signals progress through an XC9500 device, they encounter each of these delays which are tallied to arrive at a cumulative time delay for that signal. Table 1 provides a detailed definition of each parameter contained in Figure 1. The exact values of these parameters for each device can be obtained from the data sheets.

Table 1: Key XC9500 Internal Timing Parameter Definitions

Symbol	Parameter
	Buffer Delays
t_{IN}	Input buffer delay
t_{GCK}	GCK buffer delay
t_{GSR}	GSR buffer delay
t_{GTS}	GTS buffer delay
t_{OUT}	Output buffer delay
t_{EN}	Output buffer enable/disable delay
	Product Term Control Delays
t_{PTCK}	Product term clock delay
t_{PTSR}	Product term set/reset delay
t_{PTTS}	Product term 3-state delay
	Internal Register and Combinatorial Delays
t_{PDI}	Combinatorial logic propagation delay
t_{SUI}	Register setup time
t_{HI}	Register hold time
t_{COI}	Register clock to output valid time
t_{AOI}	Register async. S/R to output delay
t_{RAI}	Register async. S/R recovery before clock
t_{LOGI}	Internal logic delay
t_{LOGILP}	Internal low power logic delay
	Feedback Delays
t_F	FastCONNECT matrix feedback delay
t_{LF}	Function Block local feedback delay
	Time Adders
t_{PTA}	Incremental product term allocator delay
t_{SLEW}	Slew rate limited delay

Timing Calculation Examples

Table 2 shows how various external timing parameters are derived from the internal timing parameters. For example, t_{PD} is the sum of the input buffer time delay (t_{IN}), the logic time delay (t_{LOGI}), the flip-flop pass through delay (t_{PDI}), and the output buffer time delay (t_{OUT}), as shown in Figure 2. Note that the input buffer delay is combined with the time delay accrued when the entering signal passes through the FastCONNECT switch matrix.



$$\boxed{t_{PD} = t_{IN} + t_{LOGI} + t_{PDI} + t_{OUT}}$$

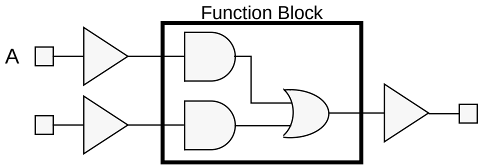
Figure 2: Simple t_{PD} Example

Table 2: Expressions for Key Timing Parameters Derived from Table 1

Symbol	Parameter	Calculation
t_{PD}	Propagation delay*	$t_{IN} + t_{LOGI} + t_{PDI} + t_{OUT}$
t_{SU}	Global clock setup time*	$t_{IN} + t_{LOGI} + t_{SUI} - t_{GCK}$
t_H	I/O hold time after GCK	$t_{GCK} + t_{HI} - t_{IN} - t_{LOGI}$
t_{CO}	Global clock-to-output*	$t_{GCK} + t_{COI} + t_{OUT}$
f_{CNT}	16-Bit counter frequency	$1/(t_{COI} + t_{LF} + t_{LOGI} + t_{SUI})$
f_{SYSTEM}	Internal system clock period*	$1/(t_{COI} + t_F + t_{LOGI} + t_{SUI})$
t_{PSU}	P-term Clock setup time*	$t_{IN} + t_{LOGI} + t_{SUI} - t_{IN} - t_{PTCK}$
t_{PH}	I/O hold time after p-term clock	$t_{IN} + t_{PTCK} + t_{HI} - t_{IN} - t_{LOGI}$
t_{PCO}	Product term clock-to-output	$t_{IN} + t_{PTCK} + t_{COI} + t_{OUT}$
t_{OE} t_{OD}	GTS to output enabled/disabled	$t_{GTS} + t_{EN}$
t_{POE} t_{POD}	P-term OE to output enabled/disabled	$t_{IN} + t_{PTTS} + t_{EN}$

* See AC Table Parameters

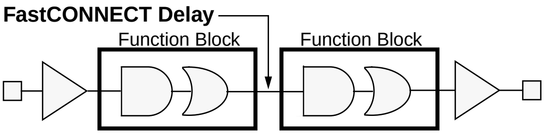
Figure 3 shows a variation on the simple t_{PD} example with the addition of cascaded product terms. The time delay from input A is slightly altered by the addition of one t_{PTA} value which accounts for the additional product terms. The XC9500 can accept and deliver product terms in either direction, so the t_{PTA} time delay will handle this factor. Also, product terms may arrive from non-adjacent macrocells, which would require an additional t_{PTA} to be added. Therefore, a single cascade delay may in some cases not be what the design software has chosen. This cascade timing can be controlled by using timing driven optimization, described in detail in the Design Optimization application note.



$$t_{PD} = t_{IN} + t_{LOGI} + t_{PTA} + t_{PDI} + t_{OUT}$$

Figure 3: t_{PD} with Cascaded P-Terms

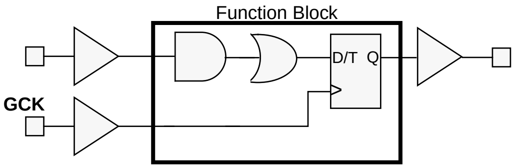
Figure 4 shows the results of supplementing single pass logic with an additional pass through another macrocell. In this case, there is a single pass through the input and output buffers, two passes through the macrocell logic, and a single pass through the feedback path. The feedback path can be either through the general feedback (t_F) or the local feedback (t_{LF}), depending on timing constraints supplied by the designer in a .CST file.



$$t_{PD} = t_{IN} + t_{LOGI} + t_{PDI} + (t_F \text{ or } t_{LF}) + t_{LOGI} + t_{PDI} + t_{OUT}$$

Figure 4: t_{PD} with Multiple Pass Logic

Figure 5 shows the situation for a simple flip-flop clocked by a global clock signal (GCK). The expressions for t_{CO}, t_H, and t_{SU} in Table 2 are valid for this arrangement.



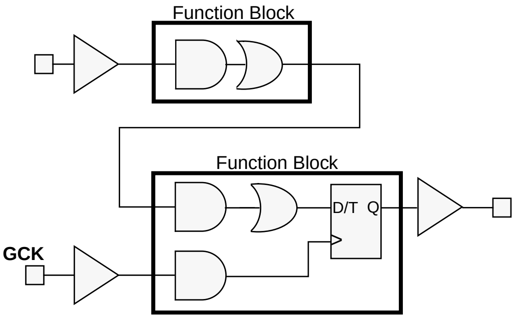
$$t_{SU} = t_{IN} + t_{LOGI} + t_{SUI} - t_{GCK}$$

$$t_{CO} = t_{GCK} + t_{COI} + t_{OUT}$$

$$t_H = t_{GCK} + t_{HI} - t_{IN} - t_{LOGI}$$

Figure 5: Simple Flip-Flop Path
(Note: Global clock)

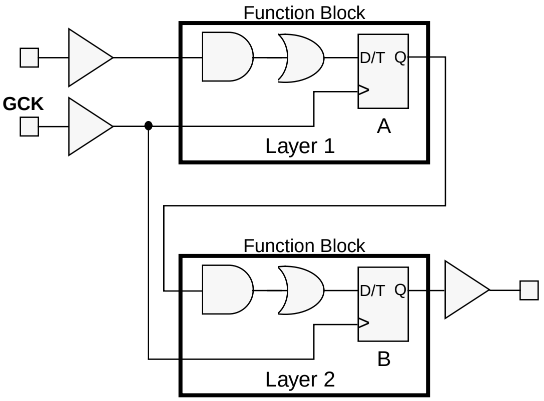
Figure 6 shows the addition of another layer of macrocell logic into the situation described in Figure 5. The t_{CO} and t_H expressions remain the same, but the t_{SU} expression is increased by another ($t_{LOGI} + t_{PDI} + (t_F$ or t_{LF})) depending on timing constraints supplied by the designer in a .CST file.



$$t_{SU} = t_{IN} + t_{LOGI} + t_{PDI} + (t_F \text{ or } t_{LF}) + t_{LOGI} + t_{SUI} - t_{GCK}$$

Figure 6: Flip-Flop with Multiple-Pass Logic
(Note: Global clock, t_{CO} and t_H are unchanged.)

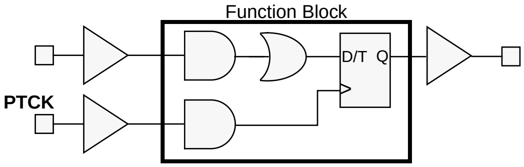
Figure 7 shows two flip-flops connected by a single level of logic, clocked by a global clock. The t_{SU} and t_H for flip-flop A are identical to that of Figure 5.



$$f_{system} = 1/(t_{COI} + (t_F \text{ or } t_{LF}) + t_{LOGI} + t_{SUI})$$

Figure 7: Multiple Flip-Flops with Single Level Logic (Note: Global Clock)

Figure 8 shows a single flip-flop with a product term clock. This arrangement differs from Figure 5 only in that the clock input comes from a product term clock. The entry for t_{PCO} in Table 2 reflects this variation. The timing for t_{PSU} and t_H is calculated using the product term clock timing parameters.



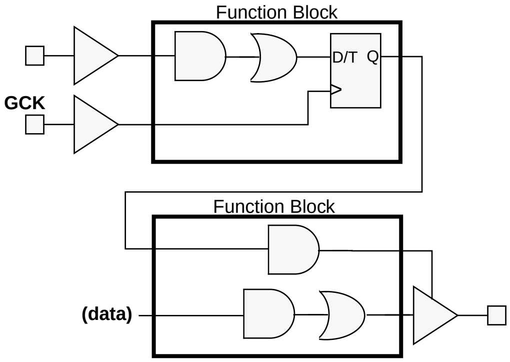
$$t_{PSU} = t_{IN} + t_{LOGI} + t_{SUI} - t_{IN} - t_{PTCK}$$

$$t_{PCO} = t_{IN} + t_{PTCK} + t_{COI} + t_{OUT}$$

$$t_H = t_{IN} + t_{PTCK} + t_{HI} - t_{IN} - t_{LOGI}$$

Figure 8: Single Flip-Flop with Product Term Clock

Figure 9 shows the timing for driving valid data onto a bus with respect to a rising clock edge, a common configuration that occurs in high speed buses. This is sometimes called t_{VAL}. In this example, it is assumed that Function Block feedback passes through the local feedback paths.



$$t_{VAL} = t_{GCK} + t_{COI} + t_{LF} + t_{PTTS} + t_{EN}$$

Figure 9: Flip-Flop-Controlled Output Enable

Conclusion

This set of examples is sufficient to describe a large number of design configurations, and other examples can easily be derived from the timing model. For manual calculations, other timing delays such as t_{SLEW} and t_{LOGILP} are easily added to the overall timing as required.