XCELL

THE NEWSLETTER FOR XILINX PROGRAMMABLE GATE ARRAY USERS

Issue 6 Fourth Quarter 1990

1990 has been an exciting year for us. After six years in business and ten successive profitable quarters, with sales doubling every year, Xilinx became a publicly held company. The Initial Public Offering was made on June 12, and at the end of this turbulent year the stock is up over 37%.

Xilinx technology is thus appreciated not only by smart designers and astute managers, but also by demanding investors who expect us to continue our rapid profitable growth. We are delighted by this confidence, and will do our best to live up to the expectations of our customers and the financial community.

If you are interested, you can find us near the end of the NASDAQ listing under "Xilinx."

Peter Alfke, editor

Table of Contents

The LCA is the Speed Champion

Of all the demands on a Programmable Gate Array, speed is the dominant requirement. Most of our customers want to implement high performance logic, often tightly coupled to a modern microprocessor with clock rates in excess of 20, or even 30 MHz.

System performance is obviously determined not only by the flip-flop toggle rate, but by the design-specific combinatorial delays between clock edges, plus the flip-flop set-up and propagation times. This **total** time must be kept shorter than a clock period, and Xilinx helps you by combining multi-level logic into **one** function generator, and thus **one** delay. Competitors with smaller logic granularity cannot achieve this performance.

• Actel uses a double multiplexer structure that is no match for the versatility of the Xilinx 4- or 5-input function generator. So the Actel user must often concatenate two or three of these modules to equal the capability of the LCA function generator. This doubles or triples the combinatorial delay and reduces performance accordingly.

• Plessey opted for an even smaller granule, one 2-input NAND gate. Up to 20 of these are required, of which up to five must be connected in series, to match the capability of our function generator. That means a long propagation delay and slow performance.

Not only are Xilinx propagation delays shorter, we actually fold one function generator delay into the flip-flop set-up time: The first function generator in front of the flip-flop D-input is free, its delay is accounted for in the set-up time.

Two additional, more subtle features enhance performance:

• On-chip clock distribution is very precise. There are two independent clock networks that can each clock every on-chip flip-flop with less than 2 ns of clock skew. Don't worry about flip-flop or latch hold times, they are zero.

• The XC3000 family can have on-chip bidirectional buses that simplify routing and avoid multiplexer delay.

In 1991, we will achieve even better performance by moving to a sub-micron process. PA

Some examples from our Design Library:

16-bit adder in 33 ns (page 6-30 of the 1989 Xilinx Data Book)
32-bit pipelined accumulator at 60 MHz (XCELL#5, page 13)
60-bit binary counter at 30 MHz (page 6-33)
40-bit loadable counter with decoded TC at 40 MHz (page 6-38)
9-bit loadable down-counter at 25 MHz (page 6-43)
8-bit loadable up/down counter at 20 MHz (page 6-34)
8-digit frequency counter at >100 MHz (page 6-44)

Component Availability (January 1991)

		44 PIN	48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		160 PIN	164 PIN	175 PIN	
		PLASTIC PLCC	PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		PC44	PD48	CD48	PC68	PG68	PC84	PG84	PQ100	CQ100	PP132	PG132	PQ160	CQ164	PP175	PG175
XC2064	-50		C	I	CI	CIM										
	-70				CI	CIM										
	-100				C	C										
XC2018	-33							MB								
	-50				CI		CI	CIMB								
	-70				CI		CI	CIMB								
	-100				C		C	C								
XC3020	-50				CI		CI	CIMB	CI	CIMB						
	-70				CI		CI	CIMB	CI	CIMB						
	-100				C		C	C	C	C						
XC3030	-50	CI			CI		CI	CIM	C							
	-70	CI			CI		CI	CIM	CI							
	-100	C			C		C	C	C							
XC3042	-50						CI	CIMB	C	CIMB	CI	CIMB				
	-70						CI	CIMB	CI	CIMB	CI	CI				
	-100						C	C	C	C	C	C				
XC3064	-50						C				CI	CIM	CI			
	-70						C				CI	CIM	CI			
	-100						C				C	C	C			
XC3090	-50						C						CI	CIMB	CI	CIMB
	-70						C						CI	CIMB	CI	CIMB
	-100						C						C	C	C	C

X1318

Current Software List

The following is a list of the current software revision levels for Xilinx's development system products. This is a listing of the diskettes currently being shipped with each of the development system products, as of January 1, 1991.

DS112 Enhanced Serial Configuration PROM Programmer ver. 3.11
XPP Program ver. 3.10

DS22-PC1 P-SILOS ver. 4.0
P-Silos ver. 4.0
XNF2SILO ver. 4.0

DS28 XACTOR ver. 3.00
XACTOR 3.00

DS290-PC1 VIEWsim ver. 3.0
VIEWsim Simulator ver. 4.0d
VIEWsim Interface and Library
XNF2WIR ver. 3.00G

DS31-PC1 Futurenet DASH Interface ver. 4.00
DASH Interface and Library
PIN2XNF ver. 2.40 and 4.00

DS310-PC1 DASH-LCA ver. 4.0
DASH-LCA Schematic Editor ver. 5.03
DASH Interface and Library
PIN2XNF ver. 2.40 and 4.00

DS343-AP1 Mentor Interface ver. 3.0
Schematic Editor and Simulator Interface ver. 3.0

DS35-PC1 OrCAD/SDT Interface ver. 2.4
OrCAD/SDT Interface and Library
PIN2XNF ver. 2.4
TTL Library ver. 1.0

DS390-PC1 VIEWdraw-LCA ver. 3.0
VIEWdraw-LCA Editor ver. 4.0d
VIEWdraw Interface and Library
WIR2XNF ver. 3.0h

DS391-PC1 VIEWlogic Interface ver. 3.0
VIEWlogic Interface and Library
WIR2XNF ver. 3.0h
XNF2WIR ver. 3.00G

DS501-PC1 XACT Development System ver. 3.00
XACT Design Editor ver. 3.00
Xilinx Design Manager ver. 1.10
Download ver. 1.00
ADI ver. 3.00 (see below)

DS501-AP1 XACT Development System ver. 3.00 on Apollo
EADI, MakeBits, MakeProm ver.3.02
XACT Design Editor on PC ver. 3.00
Download ver. 1.00

DS501-SN1 XACT Development System ver. 3.00 on SUN3
EADI, MakeBits, MakeProm ver. 3.02
XACT Design Editor on PC ver. 3.00
Download ver. 1.00

DS501-SN2 XACT Development System ver. 3.00 on SUN4
EADI, MakeBits, MakeProm ver. 3.02
XACT Design Editor on PC ver. 3.00
Download ver. 1.00

DS502-PC1 XACT 4000 Development System ver 1.0
Xilinx Design Manager ver. 2.00
PPR ver. 1.00
Makebits ver. 4.00
Makeprom ver. 4.00
LCA2XNF ver. 4.00
MEMGEN ver. 1.00
XACT Design Editor ver. 4.00
XMAKE ver. 2.01
XNFCVT ver. 4.00
XNFUPD ver. 1.00
ADI ver. 3.00 (see below)

ADI ver. 3.00
XNFDRC ver. 3.00
XNFMAP ver. 3.00
XNFMERGE ver. 3.00
MAP2LCA ver. 3.00
APR/APRLOOP ver. 3.00
LCA2XNF ver. 3.00
XNFCVT ver. 3.00

Development System Updates Improve Design Productivity

Xilinx is firmly committed to being the leading supplier of FPGA development software as well as FPGA devices. New updates to most Xilinx development system products were released this past year, as part of our ongoing commitment to improve the FPGA design tools and increase the productivity of our users. As always, these updates have been distributed to Xilinx customers with active software support contracts. Additional updates are planned for early 1991 to further enhance software capabilities and to add support for the new XC4000 family of FPGAs.

Among the most significant updates was last spring's release of version 3.0 of the Automated Design Implementation (ADI) package and the XACT Design Editor. The ADI software's improved partitioning, placement, and routing algorithms yield significant improvements in the routability and performance of automatically placed and routed LCA designs. The XACT Design Editor now supports 8- and 16-color modes, and includes several new commands. (Look for version 3.1 of the ADI package, with yet more improvements supporting the XC2000 and XC3000 family, in the second quarter of 1991!)

XACT and ADI package files that support new part/package combinations, including the XC3064PC84, XC3090PC84, and XC3090PQ160, are available on the Xilinx Bulletin Board. These package files also can be obtained by calling the Technical Support Hotline.

Version 2.4 of the DS35 OrCAD/SDT schematic-to-XNF translator was released last summer. A production release version of the DS351 XNF interface for the OrCAD/VST simulator should be released by the time this newsletter reaches you. (Those using the pre-release version that was distributed on the Bulletin Board last year should upgrade to the released version). OrCAD will be releasing version 4 of its SDT and VST tools in early 1991; Xilinx will update the interfaces as soon as possible after OrCAD ships the new versions. Interfaces and libraries that support the new XC4000 family also will be available in the spring.

The latest update to the DS31 DASH schematic interface started shipping late in December. This update includes both a new macro library that supports the XC4000 family and an additional MSI macro library for the XC2000 and XC3000 families. (The MSI macros previously had been included in a separate product called the DS311.) Versions of the DS31 are available for Sun3 and Sun4 workstations as well as the PC.

An update to DASH-LCA, the Xilinx version of the DASH editor, also began shipping in late December. This editor, now shipped in a product called the DS310, has been updated to include DASH-LCA version 5.0 from Data I/O as well as the libraries mentioned above. (Note: We discontinued the DS53 product that included the DASH-LCA editor bundled with the DS501 Design Implementation Package. Active DS53 maintenance contracts have automatically been changed to DS310 and DS501 maintenance contracts.)

DS22 updates that provide XC4000 family support for the Silos simulator also began shipping at the end of 1990.

In November, Xilinx began shipping our OEM versions of the Viewlogic Viewdraw schematic editor and Viewsim simulator, called the DS390 and DS290, respectively. These packages support XC2000 and XC3000 family design. The interfaces and libraries for both the editor and simulator are available separately in the DS391, for customers who already own the Viewlogic tools. XC4000 libraries will be added to these packages soon.

The XPP software for the DS112 Serial PROM Programmer has been updated to handle the XC1736A and XC1765 serial PROMs. Data files also are available for programming AMD's Am1736 device.

The DS343 XNF interface supporting the Mentor IDEA schematic editor and Quicksim timing simulator (under Mentor version 7.0 and Apollo OS 10.1) was released last August. Again, XC4000 family support will be added in the first half of 1991.

Complete versions of the DS501 software for the Apollo, Sun3, Sun4, and DecStation 3100 workstations are in the final stages of development, included the Design Manager, XACT Design Editor, DS112 PROM Programmer support, and download cable support. In fact, the Apollo and Sun4 versions will be "beta-testing" in January. (Versions of the Automated Design Implementation [ADI] software for these workstations are already available.)

Other new products planned for 1991 include an EDIF interface and a state machine optimization/implementation tool.

You will receive free software updates for one year after the purchase of a development system product if you mail the registration card to Xilinx Customer Service. To qualify for updates after one year, a software maintenance contract must be purchased.

As you receive new updates or newly-purchased products, please remember to fill out and return the registration cards. This will insure that you receive all relevant updates. If you have not received an update that you believe you should have gotten, or if you wish to renew an expired software maintenance agreement, please contact your local Xilinx sales representative.

BF

Rube Goldberg and the Art of LCA Design

Xilinx LCA devices bring gate-array capability to the large number of logic designers who cannot afford the cost, risk, and delay of a masked gate-array, but still want to design their own LSI circuits. More new gate-array designs are presently being implemented with Xilinx LCAs than with all other gate-array technologies combined. We love this wide acceptance of our technology, but we are also concerned about the sometimes marginal and even bad logic designs that are being implemented in our parts.

There are exciting arguments that make Xilinx LCAs your favorite choice:

"Build your design in parts, try it out, and modify it if it doesn't work!"
"Fix your mistake before your boss ever sees it!"
"Respond easily to demands created by the market or your competition!"
"Convert an idea you had in the shower to a working chip the same day!"

These are perfectly valid statements, but they do not guarantee design quality. In fact they might actually tempt the designer to be less thorough in the original approach.

"Why perform a careful analysis when I can try it out so easily?"

This attitude can lead to bad design methods and unreliable products. Here are some words of advice, based on over 30 years of systems and logic design experience, and exposure to a few hundred LCA designs over the past two years:

• Always start off with a top-down design. Look at the big picture before you implement the details. Draw a block diagram, step back, and see if you can simplify it by combining functions.

• Trade off complexity against speed. LCA circuitry is quite fast. If your clock runs much slower than 10 MHz, investigate whether you might perform the function time-multiplexed or serially.

• When you design slow logic, don't get careless. The circuitry doesn't know about your low speed; it can still react to nano-second decoding spikes on the clock or asynchronous reset lines, and might be sensitive to 10ns clock skew problems.

• Be extremely careful with asynchronous inputs. Whenever two asynchronous signals are combined, always (A-L-W-A-Y-S!) perform a thorough worst-case analysis of what happens under the most extreme phase relationships between those inputs.

Use the combination of two asynchronous signals to affect **only one** flip-flop, either to clock it, to reset it, or to synchronize the two signals. One flip-flop will either react or not react to a marginal signal; but several flip-flops might disagree and may cause your system to crash.

Remember, according to Murphy's Law, if something bad **can** happen, it **will** happen, and usually at the most inappropriate moment.

• Never use a decoder to clock or reset a flip-flop or latch asynchronously. Uncontrollable decoding spikes may cause unpredictable behavior that may be temperature-, voltage- or lot-dependent. A classical example: Using the Terminal Count output from a 74161 as a clock is an invitation to disaster.

• Be aware of the metastability problem, but don't be paranoid about it. Read pages 6-20 and -21 of our 1989 Data Book; it shows that an extra 10 ns of tolerable propagation delay can reduce the metastability problem to statistical insignificance.

• Use the global clock distribution network which eliminates all clock skew problems. If you have to distribute a clock signal through general-purpose interconnect and "magic boxes" to several flip flops, always check for clock-skew problems, even if your design is otherwise not time-critical. There usually is an easy cure: Run the clock delays in a direction opposite to the data flow. Clock the most significant part of a counter early, the less significant part later, and all clock skew problems disappear.

• If you have used nonsynchronous logic tricks, analyze them very carefully. Check for potential problems with faster parts. Evaluate your design in the fastest LCA that you can buy (presently the -100), to check for potential future problems. There are also two simple ways to change the delay in LCAs mounted on your board: Just vary the temperature or the supply voltage: Heat makes CMOS slow, cold makes it fast; low Vcc makes it slow, high Vcc makes it fast.

If your design fails at high temperature and/or low Vcc, then you are just pushing performance and are running into problems with excessive propagation delays. You might want to improve the routing or use a faster part.

If, however, your design fails at low temperature and/or high Vcc, you have reasons to worry. Take a deep breath and analyze your design for asynchronous abnormalities and for clock skew problems. If you don't fix these problems immediately, they will bite you in the future. CMOS processes are constantly being improved and shrunk. Circuits will get faster, and what was an innocent glitch in a slow part may jeopardize your system in the future.

Logic design is both an art and a science. There are elegant designs and there are kludges; there are rugged designs and there are flimsy contraptions that will inevitably fail sooner or later.

Standard LSI circuits are crafted by experienced logic designers who know what's at stake, are aware of the possible pitfalls and know how to avoid them. And, they simulate their designs and take the time to get it right.

Programmable gate arrays give the user full responsibility for every aspect of the logic design. We hope that the user community is up to this challenge.

The cartoonist Reuben Lucius Goldberg (1883 - 1970) was known for his whimsical drawings of ludicrously intricate machinery meant to perform simple tasks.

PA

Using ADI 3.0 Effectively

The May 1990 release of the automatic design implementation software is a significant enhancement over previous versions. It achieves dramatic improvements in terms of both routability and system performance.

We have, however, also received calls from some users saying that they fail to see any improvement, or that the old software gave better results.These problems arise when users over-constrain their design or otherwise limit the software.

To get the best results from the new software, follow these three rules:

• **Don't use CLBMAPs unnecessarily.**
With older versions of the software it was sometimes necessary to manually map logic such as registers or counters to ensure that a certain order prevailed, e.g. that bits 0 and 1 were mapped together. With bus lines labeled properly, XNFMAP V3.0 will do this automatically. In addition, this version uses better algorithms to combine logic, and from what we've seen, it does as good a job as you would do manually.

• **Don't use constraint flags.**
Net constraints are the biggest reason why designs do not route completely. To make sure that critical timing paths are met, some designers flag every net in the path as critical. Because of this, APR distorts the placement by putting the critical blocks next to each other, instead of placing them where they should be relative to the rest of the design.

Once the blocks are poorly placed, routing is more difficult. And since the blocks are packed so tightly, longer routing paths are required to go around congested areas to get to the destination pins. When too many flags are put in a constraint file, designs end up running slower than they would have if no flags were used at all!

• **Don't place CLBs.**
It's best to let APR choose initial block locations. After the design has been placed, you can use XACT to optimize the placement.

What has been a long list of DOs and DON'Ts in previous articles has been shortened to just these three DON'Ts.

The tedious steps of mapping logic, floorplanning, and weighting nets, which were helpful or even necessary for our old software, are now an obstacle for ADI 3.0 to overcome.

So make your job and our software's job easier: skip those steps.

We in Xilinx Applications have been using ADI 3.0 for over a year, starting with early, pre-alpha release versions. We have been able to complete numerous real designs that could not be completed with the previous revisions of our software. Based on this experience, we have developed a methodology for routing tough designs, which is described on the following page.

BON

ADI 3.0 Methodology for Tough Designs

1. **Run XNFMAP** with the -M option chosen. This tells the program to ignore CLBMAPs.
2. **Run MAP2LCA** without any command line options.
3. **Delete the .SCP file.** (If IO locations have been specified, keep them but remove everything else.)
4. **Run APR** without any command line options chosen.
5. **Load the design in XACT** and check path delays.
6. If necessary, **weight a few of the nets**, and run APR with the -L option. This will keep the placement locked and re-route the design. This will not take long, since the placement phase is skipped.
7. If the design still isn't routing, there are four ways to solve this problem, depending on the nature of the design:

a) Full Designs

If the CLB utilization of a chip is too high, the design may have trouble routing.

Signs:

- More than 85% of the chip is used.
- Many large net delays.
- Many unrouted pins and nets.

Solutions:

- Run XNFMAP using -CSI options.
- Place blocks and pre-route nets.
- Use a larger part.

b) Over-constrained Designs

Designs with locked IO blocks, many net constraints, or CLBMAPs can be hard to route.

Signs:

- IO assignments in schematics.
- Many net-weight flags in schematics.
- CLBMAP symbols in schematic.

Solutions:

- Run XNFMAP using -M option (ignore CLBMAPs)
- Edit .SCP file to remove constraints and re-run APR.

c) Structured Designs

Full designs with many registers and buses can be hard to route.

Signs:

- Check to see if registers are lined up vertically.
- Check to see if buses are laid out in a logical order.

Solutions:

- Manually line up registers using XACT and re-run APR using the -L option (lock blocks, just route) (Figure 1).

d) Sparse Designs

Designs which use relatively few CLBs (less than 50%) sometimes don't route.

Sign: A partially utilized device won't route.

Solution: In a constraint file, prohibit a few columns in the middle of the chip (Figure 2).

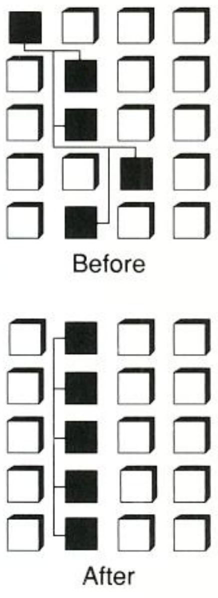


Figure 1

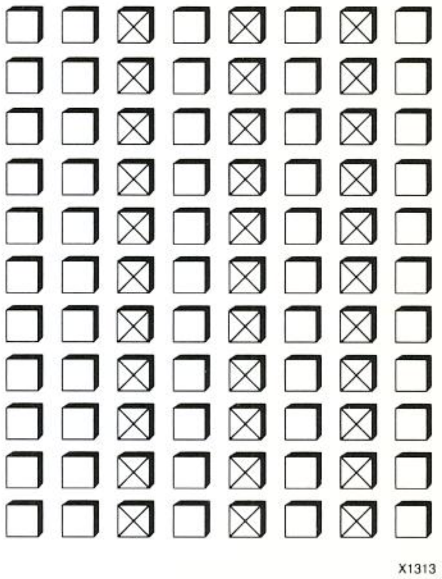


Figure 2

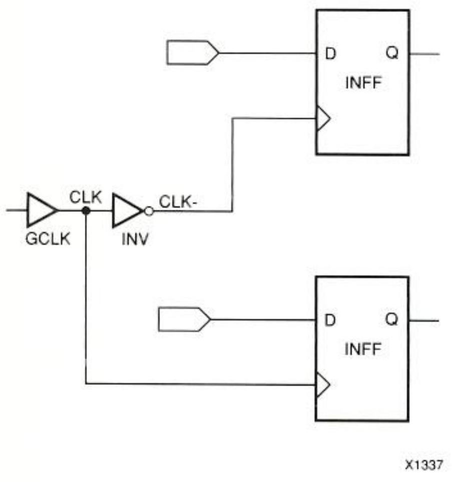
Dual Phase Clocking for IOBs

Some XC3000 designs use IOB storage elements with mixed clock polarity (see below). The chip is capable of doing this, but APR does not support dual phase clocking from either the GCLK or the ACLK net. APR aborts with Error Message 113:

"Both phases of clock net GCLK (ACLK) are used for clocking IOB storage elements. Since APR does not support dual phase clocking of IOB storage elements by a single net, all IOBs with storage elements clocked by net GCLK (ACLK) must be locked in such a way that any IOBs that use opposite phases of the clock do not lie along the same edge of the LCA".

The clock signal cannot be inverted inside the IOBl, and APR will not use two separate nets to supply the regular and inverted sense clock signals.

APR can only use one clock polarity per die edge. This means that the user would have to lock all pins in order to prevent placement of IOBs triggered with mixed polarity on the same die edge. Such locking of I/O Blocks, however, is very undesirable because it creates routing difficulties.



Work-Around:

Using XACT, it is possible to route the ACLK (but not the GCLK) net to separate I/O clock lines on a single die edge, feeding both clock polarities to the IOBs.

The following steps must be performed:

1. Change the schematic so that the IOBs are clocked with the same ACLK polarity. Make a note that the schematic no longer represents the intended circuit behavior.
2. Route your design with APR.
3. Go into XACT EditLCA and modify the IOBs for dual phase clocking: invert the clock polarity inside the IOB.
   Note: In EditBlock this inversion appears to be inside the IOB. Physically, it will be implemented with an inverter in the clock line.
4. Now split up the clock signal into two opposite polarity signals. In EditNet, turn off the PIPs that connect the modified IOB clock inputs to the clock line assigned by APR and turn on the PIPs that connect it to the second line. Ignore all warnings that DRC will give you.
5. Use the 'Route' command to complete the inverted net. This will automatically turn on a hidden inverter, since the router sees that certain IOB flip flop clock inputs are inverted.
6. Run DRC to check your modifications.
7. Check the net delay with the 'Delay' command. Note that there is some skew between the noninverted and the inverted clock signals.

Remember, since you edited the LCA, your schematic does not represent the final circuit. The same dual phase clocking method can also be used to enable latches.

ADI Performance on Different Platforms

We recently ran a large number of real designs on a variety of PCs and workstations. Here is a list of their normalized performance, relative to the '386.

(Performance is the inverse of execution time running ADI 3.0)

PC '386, 20 MHz	**1.0**
PC '486, 25 MHz	**2.8**
Sun 3/80	**1.5**
SPARCstation1+	**6.0**
DEC DS3100	**5.5**

The '486 has really moved the venerable PC into the workstation arena, and that's even before we are running ADI in '386 native mode!

LCA Drives LCD Directly

Non-multiplexed Liquid Crystal Displays (LCDs) must be driven with an ac voltage of 30 to 100 Hz and 10 V peak-to-peak amplitude, **without any dc component**. Generating this voltage is surprisingly simple in an LCA, using only half a CLB plus one IOB per segment.

The back plane of the display is driven by a low frequency (100 Hz) square wave BP, oscillating between 0 and +5 V, and this signal is also used to control the inverting/non-inverting of Data. When Dout is **in phase** with BP, there is no ac-voltage across the segment, and it looks transparent. When Dout is in **counter-phase** with BP, there is an ac-voltage across the segment, and it appears dark = on.

An additional Light Blanking Input (LBI) can force data to be blank=zero, useful for leading-zero suppression. NJC

Worst-Case Input Set-up Time

Timing parameters in programmable devices are more difficult to specify than in fixed-program devices, because the user can affect some parameters through routing.

Inside the LCA, a synchronous design is easy to analyze, because hold time is not an issue, since clock skew is much shorter than the minimum clock-to-Q delay of any CLB. The only concern is for performance: Is the sum of propagation delay and set-up time less than the clock period?

The set-up time at the LCA **input** is more complex, since the clock delay from the clock pad to the internal clock cannot be ignored.

The data sheet specifies the IOB set-up time with respect to its clock (**not with respect to the clock pad!**). The unavoidable delay from clock pad to internal clock must obviously be **subtracted** from the specified set-up time, to arrive at the system set-up time.

What is the maximum value for the input set-up time, and what is its minimum value? Is there a risk for a hold-time requirement?

Maximum Set-up Time

The longest input pad set-up time, the one that determines system performance, is the specified longest IOB flip-flop set-up time minus the shortest clock delay that is consistent with such a long set-up time.

The question is: How well do such delays track?

Here is an **unrealistic** assumption:
"All delays track perfectly. In a given part, at any given temperature and supply voltage, the ratio of any actual parameter value to its specified worst case value is the same constant. "

If this were true, the max set-up time would simply be the difference of the two specified max values for flip-flop set-up time and clock delay.

Here is another **unrealistic** assumption:
"There is no delay tracking. Any parameter can vary between its max and min value, independent of all other parameters."

If this were true, the max system set-up time would be the difference between the specified max flip-flop set-up time and the minimum clock delay, whatever small value that might be.

Both these assumptions are wrong.

The circuits being evaluated reside on one piece of silicon. They were processed together, and they have a common temperature and supply voltage. All delay parameters will, therefore, track reasonably well. But since all parameters do not necessarily depend on the same physical phenomena (resistance, capa-citance, threshold voltage etc.) in the same way, they will not track perfectly.

We make the assumption that tracking in any one device will be better than 70%.

All ratios of actual delay to specified worst-case delay for all parameters on the same device at any instant will be within a two-to-one range.

Set-Up and Hold Times

Set-up time describes the requirement for valid input data **prior to** the clock edge.

Hold time describes the requirement for valid input data **after** the active clock edge.

Any particular flip-flop at a particular temperature and supply voltage clocks in the data that happens to be at its input during an extremely narrow picosecond timing window. (If data changes during this narrow window, the flip-flop goes metastable). The width of this window is constant, but its position varies, depending on processing, temperature and Vcc.

The longest set-up time describes the earliest possible position for this window; the longest hold time describes its latest possible position. If no hold time is specified, the set-up time will always be positive, i.e. the window will always be before the clock edge.

These critical set-up and hold time values are often listed in the min column of the data sheet, conforming to an ill-conceived convention established in early 7400 data sheets. Enlightened people have argued for decades that these are really max limits of device parameters, but it has become senseless to fight over form when (hopefully) everybody agrees on the meaning.

• If one delay is close to the specified max value, then all the others will be between 70% and 100% of their respective max values.

• If the relatively slowest parameter is at 50% of its specified max value, then all the other parameters will be between 35% and 70% of their respective max values, etc. (The user should feel safe with this conservative assumption. In reality, parameters track much better than this.)

The longest set-up time is, therefore, the specified max IOB flip-flop set-up time minus 70% of the specified max clock delay.

Example: XC3020-100 using CMOS compatible TCLK input:

Ts max= 17 ns -0.7 •7 ns = **12.1 ns**

Minimum Set-Up Time and Possible Hold Time Requirement

The shortest possible set-up time is the minimum IOB set-up time minus the longest value for the clock delay that is consistent with such a short set-up time.

The minimum value for the flip-flop set-up time is not specified, since it is not readily testable. A very conservative guess puts it as short as 10% of the specified max. value. This can only occur at low temperature and high Vcc.

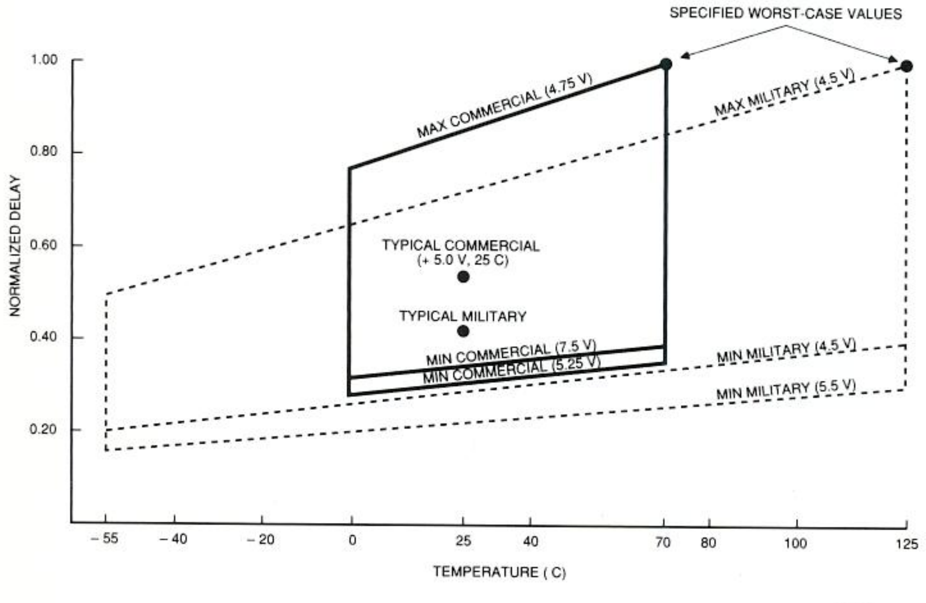
In line with the previous discussion about tracking, the maximum clock delay might then be as long as 14% of its specified max value.

Example: XC3020-100 using CMOS compatible TCLK input:

Ts min= 0.1•17 ns - 0.14•7 ns = **0.7 ns**

This means that the data-to-clock set-up time window on the LCA inputs (pads) is always somewhere between 12.1 ns and 0.7 ns. This is a wide range, but the value is always positive.

There is no hold time requirement. Data may change simultaneously with the clock, provided the clock drives the CMOS-compatible LCA input and uses the Global or Alternate clock



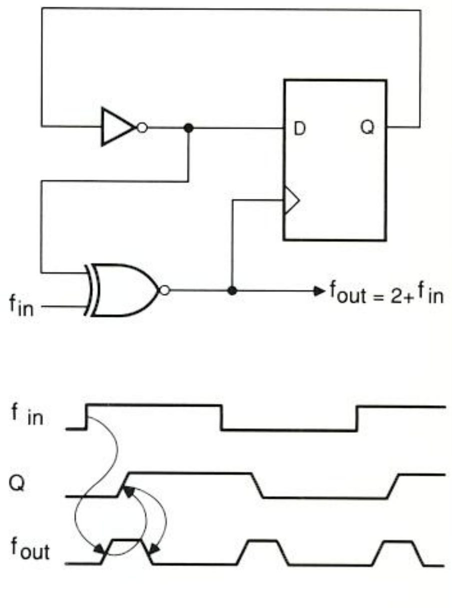
X1045

Double the Clock Frequency

A 50% duty cycle input can be doubled in frequency, provided the resulting 2f clock can tolerate a wide variation in duty cycle. The circuit below generates an output pulse in response to each transition on the input.

The output rising edge is delayed one TILO from either input transition. The output High time is the sum of a clock-to-Q delay plus two TILO delays, about 25 ns in a fast part. This output pulse will clock other flip-flops on the same die reliably. (At low temperature and high Vcc the pulse will be shorter, but the flip-flop response is also faster under these conditions)

PA



X1312

Deciphering Setup and Hold Time

Anyone looking at the Xilinx netlist format file (commonly referred to as the **XNF** file) may be surprised to find that the setup and hold times are different from the values specified in the data book. This difference is due to the fact that the XNF models use different reference points in specifying setup and hold time requirements.

The table below compares the timing parameters as seen in the data book versus those seen in the XNF file. (These are parameters for XC3000-70 devices.)

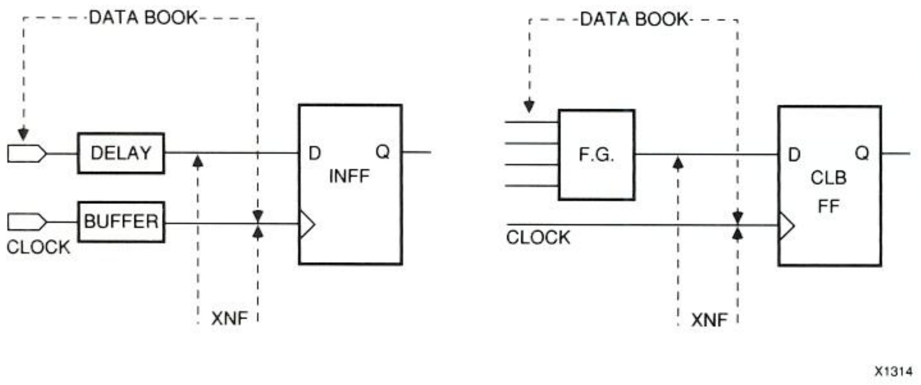
The setup time for INLAT and INFF specified in the data book is the input at the pad to the clock at the flip-flop; while the timing shown in the XNF file is from the input and at the FF to the clk. Therefore the timing window in the XNF file is shifted by 17 ns because the delay from pad to flip-flop input is 17 ns.

For the CLB flip-flop, when a,b,c,d, and e pins are used, the data book describes the timing as between inputs at the CLB pins and the clock at the flip-flop. However, the XNF file shows the setup and hold requirement with respect to the data **after** the function generator. Again, the delay time through the function generator is 7 ns, thus shifting the xnf timing by 7 ns.

In summary, there is no contradiction between XNF and the data book–they just use different reference points.

JT

	data book		xnf file	
	setup (ns)	hold (ns)	setup (ns)	hold (ns)
INLAT	**20**	**0**	**3**	**17**
INFF	**20**	**0**	**3**	**17**
OUTFF	10	0	10	0
CLB F/F: di is used	5	4	5	4
a,b,c,d,e are used	**8**	**0**	**1**	**7**



XC3090 in 84-Pin PLCC

The XC3090, largest member of the XC3000 family, is now available in the space-saving 84-pin PLCC package. Since the XC3090 requires two additional Vcc and Ground pins, the resulting pin-out differs slightly from the other XC3000 devices in this package.

Two additional Vcc connections on pins **2** and **42**
Two additional Ground connections on pins **21** and **65**
INIT-I/O moved from 42 to **41**
D3-I/O moved from 65 to **66**
CS1-I/O moved from 66 to **67**
D2-I/O moved from 67 to **68**
A13-I/O moved from 2 to **3**
A6-I/O moved from 3 to **4**
A12-I/O moved from 4 to **5**
A7-I/O moved from 5 to **6**

XACTOR Readback Error

While executing a Readback command, XACTOR2 will sometimers give the error message:

```
Readback failed to find pad bits
```

We have identified the following possible causes for this error:

1. A bug in XACTOR2 version 2.10, where several data files were not made properly. This has been fixed in version 3.00, which all users with a valid update contract will receive.

2. The part type on the XACTOR2 screen differs from the real part type on the circuit board. Make sure that the part types match.

3. An outside signal has held CCLK Low for more than 5 µs, thus violating the part specification.

Errata

Longer Write Enable in XC2000 Parts

XCELL #5 page 11 mentions that the peripheral mode write enable pulse can be as short as 50 ns and still work reliably worst case.

That is true for the XC3000 family, but not for the XC2000 devices, which may require >250 ns. Sorry for the oversight.

PA

I/Os Float High During Configuration

The comment under "Unrestricted User I/O Pins" on pages 2-31 and 2-79 of the 1989 Data Book is misleading: **Each unrestricted user I/O pin** (plus the special pins mentioned on the follwing page) **has a weak pull-up resistor** of 40 to 100 kΩ that becomes active as soon as the device powers up, and stays active until the end of configuration.

The pull-up resistor values on page 6-14 are correct, but the comment below, that the values increase at a logic High, is wrong. These pull-up "resistors" are actually p-channel transistors. Their resistor value is actually lower at a logic High.

The first lines of page 6-19 are misleading; they should read: "During configuration, all I/O pins not used for configuration are 3-stated **with a weak pull-up resistor of 40 to 150 kΩ and all latches are held reset until the chip goes active."**

PA

Presetting Flip-Flops In SILOS

In simulation, there may be times when we want to have the state of a flip-flop at a value other than the default 0. (For example, we want to start a counter at EF instead of at 00.)

The SET command in SILOS can be used to temporarily set the flip-flop output to a specific state before the next clock shifts in new data. To do this, we need to set the node directly connected to the output of the flip-flop. The output signal seen on the XACT editor screen or in the XNF file usually is not the immediate output node of the flip-flop. The following steps will help in determining the proper node to set:

A. For CLB flip-flops:

1. Begin with the name of the flip-flop output net
2. For a timing simulation, add .QX or .QY to specify the CLB output pin on which the net is located; for a logic simulation, add neither
3. Add a prefix of (and a suffix of (QB
4. Invert the polarity of the set value

For example, if the signal FRED is attached to the X output of a CLB, and we want FRED = 1, then the correct way to set it would be:

```
SET (FRED.QX(QB = 0
```

B. For INFF or OUTFF:

1. Begin with the name of the INFF or OUTFF output net
2. Add a prefix of (and a suffix of (QB
3. Invert the polarity of the set value

For example, if the signal SAM is attached to the Q output of an INFF,and we want SAM = 1, then the correct way to set it would be:

```
SET (SAM(QB = 0
```

C. For INLAT:

1. Begin with the name of the INLAT output net.
2. Add a prefix of (and a suffix of (M

For example, if the signal BOB is attached to the Q output of an INLAT, and we want BOB = 1, then the correct way to set it would be:

```
SET (BOB(M = 1
```

JT

The Effect of Marginal Supply Voltage

Since Xilinx LCAs store their configuration in static latches, some users have asked about the integrity of the configuration program under abnormal supply voltage conditions.

Here is a complete description of device behavior during supply ramp-up and ramp-down. Since XC3000 and XC2000 devices behave differently, they are described separately.

XC3000 Family

When Vcc is first applied and is still below about 3 V, the device wakes up in the pre-initialization mode (see fig. 18 on page 2-15 of the Xilinx Data Book). HDC is High; INIT, LDC and D/P are Low, and all other outputs are 3-stated with a weak pull-up resistor.

When Vcc has risen to a value above ~3 V, and a 1 and a 0 have been successfully written into two special cells in the configuration memory, the initialization power-on time delay is started. This delay compensates for differences in Vcc detect threshold and internal CCLK oscillator frequency between different devices in a daisy-chain. The initialization delay counts clock periods of an on-chip oscillator (CCLK) which has a 3:1 frequency range depending on processing, voltage and temperature. Time-out, therefore, takes between 11 and 33 ms for a slave device, four times longer for a master device. This factor of four makes sure that even the fastest master will always take longer than any slave. We assume that the worst case difference between 33 ms and 4x11 ms is enough to compensate for the Vcc rise time spent between the threshold differences (max 2 V) of devices in a daisy chain. Only in cases of very slow Vcc rise time (>25 ms), must the user hold RESET Low until Vcc has reached a proper level.

After the end of the initialization time-out, each device clears its configuration memory in a fraction of a millisecond, then tests for inactive RESET, stores the MODE inputs and starts the configuration process, as described on pages 2-15 through 2-21 of our Data Book.

After the device is configured, Vcc may dip to about 3.5 V without any significant consequences beyond an increase in delays (circuit speed is proportional to Vcc), and a reduction in output drive.

If Vcc drops into the 3-V range, it triggers a sensor that forces the device back to the pre-initialization mode described above. All flip-flops are reset, HDC goes High; INIT, LDC and D/P go Low, and all other outputs are 3-stated with a weak resistive pull-up. If Vcc dips substantially lower, the active outputs become weaker, but the device stays in this pre-initialization mode. When Vcc rises again, a normal configuration process is initiated, as described above.

The user need not be concerned about power supply dips: The XC3000 family LCA stays configured for small dips, and is “smart enough” to reconfigure itself (if it is a master) or to ask for reconfiguration by pulling INIT and D/P Low (if it is a slave). An XC3000 device will not lock up; the user can initiate re-configuration at any time just by pulling D/P Low or, if D/P is Low, by forcing a High-to-Low transition on RESET .

XC2000 Family date code trailing letter D or E (current production)

When Vcc is first applied and is still below about 3 V, the device wakes up in the pre-initialization mode (see fig. 12 on page 2-67 of the Xilinx Data Book). HDC is High; LDC and D/P are Low, and all other outputs are 3-stated.

When Vcc has risen to a value above ~3 V, and a 0 has been successfully written into a special cell in the configuration memory, the initialization power-on time delay is started compensate for differences in Vcc detect threshold between different devices in a daisy chain. The initialization delay counts clock periods of an on-chip oscillator (CCLK), which has a 3:1 frequency range depending on processing, voltage and temperature. Time-out, therefore, takes between 11 and 33 ms for a slave device, four times longer for a master device. This factor of four makes sure that even the fastest master will always be slower than any slave. We assume that the difference between 33 ms and 4x11 ms is enough to compensate for the Vcc rise time spent between the threshold differences (max 2 V) of devices in a daisy chain. In cases of very slow Vcc rise time (>25 ms), the user should wait until Vcc has reached a proper level and then initiate reconfiguration by means of a High-to-Low transition on RESET, as described in more detail below.

After the end of the initialization time-out, each device clears its configuration memory in a fraction of a millisecond, then tests for inactive RESET, stores the MODE inputs and starts the configuration process, as described on pages 2-66 through 2-71 of the 1989 Data Book.

After the device is configured, Vcc is allowed to dip to about 2 V without any significant consequences beyond an increase in delays (circuit speed is proportional to Vcc), and a reduction in output drive.

If Vcc drops below 2 V, but does not dip all the way below 0.1 V, the device might lose its stored information in such a way that it will not recognize a valid Vcc level.

To force re-configuration, the user must detect valid Vcc, then force a **High-to-Low transition on RESET while D/P is Low.** RESET must first be High for at least 100 ns, then be forced Low for at least 6 µs, and then be made High again for device configuration. This requirement is described in the left hand bottom paragraph on page 2-72 of the 1989 Data Book.

XC2000 Family date code trailing letter B or C, or no letter

Early XC2000 devices did not include the circuitry that responds to the High-to-Low transition on RESET, as described above. Once Vcc has dropped below 2 V, these devices may require a very low level, below 0.1 V, on Vcc before they can be re-configured.

PA

Powerdown Might Change Mode Settings

The PWRDWN pin should be High whenever the device tests the Mode pins prior to configuration. A Low on PWRDWN not only isolates all chip inputs (except RESET) from the internal logic, but also makes them appear as unconditional Highs to the internal logic.

A Low on PWRDWN will, therefore, test the mode signals as 111 (Slave Mode) irrespective of the actual levels applied to the mode pins. This will prevent configuration in any mode except Slave mode.

The safest solution is to have PWRDWN always High during the whole Initialization, Clear, and Configuration process. This lasts only a small fraction of a second, and draws insignificant power, since all outputs are 3-stated.

PA

Consultants Can Help You

We will occasionally publish names of consultants who are known to be able to assist you with your LCA design. Here is the first one, a company that we really can recommend since we know many of their satisfied clients:

HighGate Design
12380 Saratoga/Sunnyvale Rd.
Suite 8
Saratoga, CA 95070.
Phone: (408) 255 7160
Fax: (408) 255 7162.

They do:

Evaluation of LCA suitability

Conversion of logic into efficient LCA structures

Partitioning of functions outside and inside (multiple) LCAs

Routing of high-density logic and fast state machines

Simulation of time-critical functions

Training your designers on Xilinx development systems.

PA

Fast Clock Output from XC3000

Some designs need to output the LCA-internal clock with a minimum of delay. There are six such direct output pips, three for GCLK, and three for ACLK.

GCLK can be routed to the left- and right-most pads on the top of the die, and to the top pad on the left side of the die.

At present, such routing cannot be specified in the schematic, but must be done in XACT.

ACLK can be routed to the left- and right-most pads at the bottom of the die, and to the bottom pad on the right edge of the die.

With a few nanoseconds of extra delay, **ACLK** (but **not** GCLK!) can also be connected to the left one out of every pair of pads on the top or bottom edge of the die, and, with an even longer delay, can be routed to other pads through General Interconnect.

Any XNFDRC warning #502 does not apply to this situation, and can safely be ignored.

TCW

Single Line Switch Debouncer

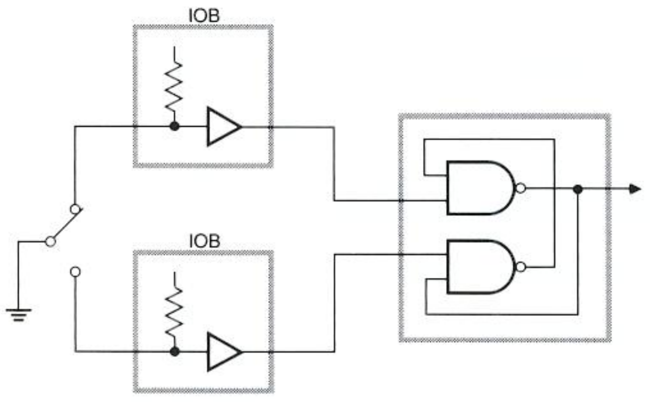
Wherever a switch drives the logic in the LCA directly, it is advisable to debounce the switch signal. And if the switch drives a clock input, debouncing is absolutely mandatory. A switch inevitably makes and breaks contact in fast succession at millisecond intervals, whenever it is being closed, sometimes even when it is being opened. To the nanosecond logic in the LCA, such switch bounces look like repetitive switch operations.

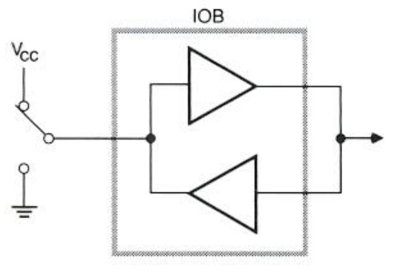
The classical switch debouncer uses two inputs to a latch. This circuit requires two IOBs and one CLB.

There is a much simpler circuit that performs the same function, using only one IOB, with the output permanently enabled and in fast mode (not slew-rate limited). Whenever the switch is being activated and reaches the new contact position, it short-circuits the IOB output to the new supply rail. This causes a momentary (<10 ns) current of 50 to 100 mA that does no harm, but might require some attention to power-supply decoupling.

If anybody is concerned about this jolt, don't worry! The same current spike flows whenever the IOB switches a load of a few hundred picofarads. Such operation is usually repeated much faster than a mechanical switch can be operated.

Debouncing a single-pole switch is a far more complicated task. It inevitably requires some kind of timing element (one-shot, counter) to discriminate between fast bouncing and deliberate repetitive operation. (kHz vs Hz). That's why the PC uses a dedicated microcontroller for the keyboard.





X1315

Download Cable Configures Daisy Chain

In the distant past, before ADI 3.0, the download cable could not configure a daisy chain of devices. Now, the Download program V1.00, part of the ADI 3.0 update, creates a bitstream even for daisy-chained devices.

We have also been asked whether a simple PC **without a software protection key** can be used with the download cable. Many users have a sophisticated PC for software development in the office, but a simpler PC in the hardware lab. No problem:

1. Use XACT to make a .bit file for each chip.
2. Use MAKEPROM, part of XACT, to make a single PROM file. MAKEPROM can make a file for a sigle device or a daisy chain.
3. Go to the other PC (It does **not** need a key!). Connect the download cable and use the download program V1.0.
4. Type: DOWNLOAD, followed by the PROMfile name, followed by <ENTER>. PA

Park CCLK High

The CCLK pin must not be held Low for more than 5 µs. Dynamic circuitry inside the LCA can reach an unknown state when capacitors lose charge during excessive CCLK Low time. If CCLK is held Low after configuration, a subsequent Readback may not function properly, reading back wrong information. Make sure that CCLK has been parked High for several milliseconds before the beginning of Readback. TCW

XC3000 Output Voltage/Current Characteristics

Figure 1 shows the current sink capability of the n-channel output transistor. At V_{OH} <4 V, the transistor behaves like a 25 Ω resistor. Above 4 V, the transistor becomes saturated and sinks a constant current of about 100 mA. Above 5.7 V, the clamp diode sinks additional current into $V_{CC.}$ If this current exceeds 100 mA for several microseconds, it can lead to latch-up which is usually destructive.

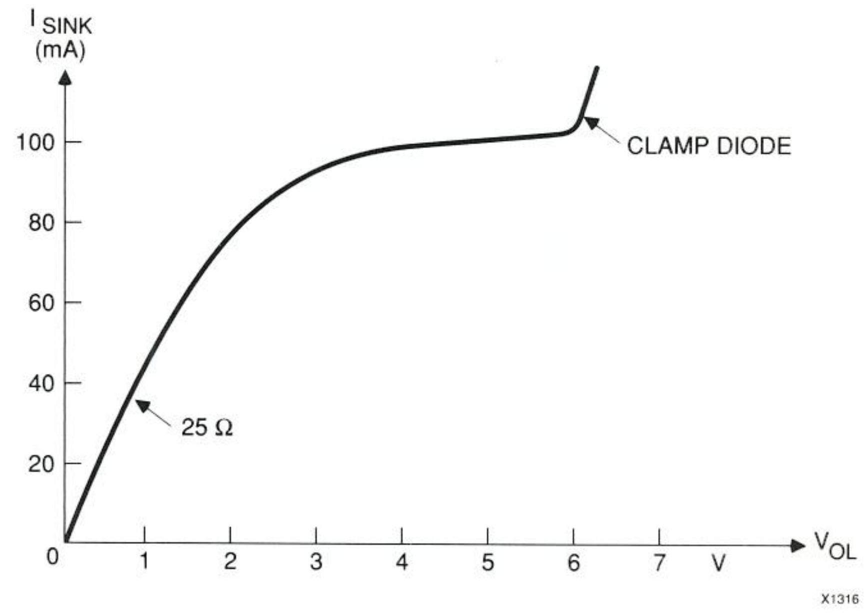


Figure 1

Figure 2 shows the current source capability of the p-channel output transistor. At V_{OH} >1 V, the transistor behaves like a 65 Ω resistor. Below 1 V, the transistor shows some saturation, and sinks about 60 mA. When the output is being driven more than 0.7 V negative with respect to Ground, the clamp diode sinks additional current into Ground.

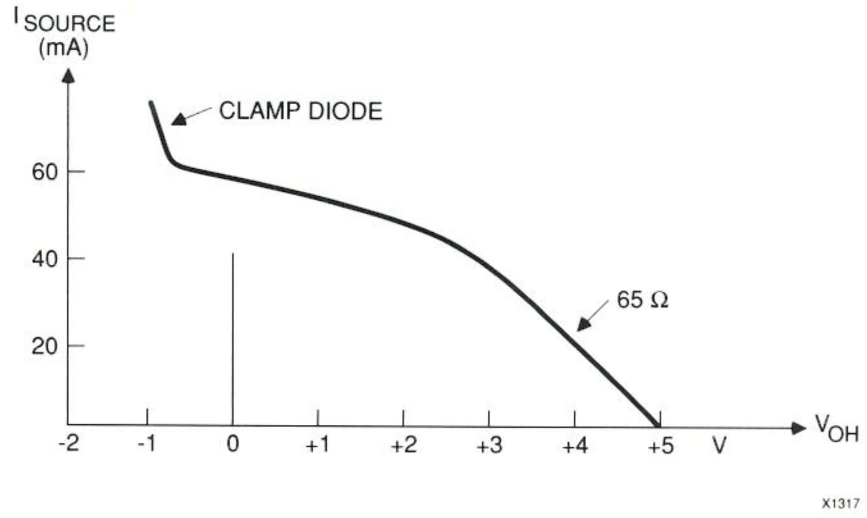


Figure 2

These I-V characteristics are based on measuring typical devices at VCC=5 V and room temperature. The current is less at hot, and higher at cold temperatures.

Xilinx does not recommend a continuous output or clamp current in excess of 20 mA.

PA

Package Weight and Thermal Impedance

Lead Count	Package Type	Grams	Theta J-A
8	Plastic Dual-Inline Package	0.52	
8	Ceramic Side-Brazed Package	0.95	110
20	Plastic Leaded Chip Carrier	0.70	79
44	Plastic Leaded Chip Carrier	1.20	44
48	Plastic Dual-Inline Package	7.90	
48	Ceramic Side-Brazed Package	8.0	
68	Plastic Leaded Chip Carrier	4.8	41
68	Ceramic Pin-Grid Array	6.95	40
84	Plastic Leaded Chip Carrier	6.80	29 to37
84	Ceramic Pin-Grid Array	7.25	30 to 37
100	Plastic Quad Flat Package	1.60	58 to 75
100	Ceramic Quad Flat Package	3.60	58 to 75
132	Plastic Pin-Grid Array	8.10	
132	Ceramic Pin-Grid Array	11.75	24 to 27
160	Plastic Quad Flat Package	5.80	
164	Ceramic Quad Flat Package	8.35	33
175	Plastic Pin-Grid Array	10.60	22
175	Ceramic Pin-Grid Array	28.40	17

XILINX

2100 Logic Drive
San Jose, CA 95124-3450

BULK RATE
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 2196