XCELL

THE NEWSLETTER FOR XILINX PROGRAMMABLE GATE ARRAY USERS

Issue 2 First Quarter 1989

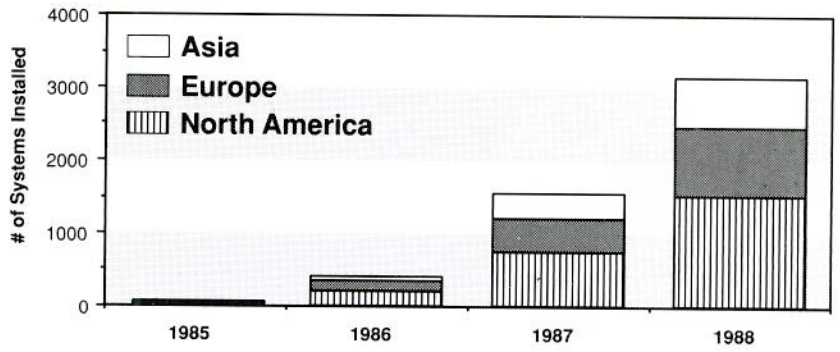
This newsletter is sent automatically to every registered Xilinx system owner. If any other user of our devices wants to be put on the mailing list, just drop us a note.

We want to reach as many designers as possible. We want this to be a widely read source of technical information, not a promotional mailer. Questions, suggestions and even critical comments are always welcome. Nothing is so good that it cannot be improved.

Peter Alfke, Editor

Table of Contents

XILINX Development Systems Installed By Year-End



1988 Was a Great Year for Xilinx

In 1988 Xilinx grew dramatically; sales almost tripled from 1987. While other High Tech companies began to experience market softness, Xilinx continued to grow much faster than the market. While other semiconductor manufacturers layed off people, we hired. While they shut down, we worked overtime.

1988 was the year of the XC3000 family. Production volume shipments of the XC3020 and XC3090 began in early 1988. Total shipments of Xilinx Programmable Gate Arrays passed the one-million unit mark. In the second half of the year we sampled the XC3030 and 3042. These two products are moving towards production, and samples of the XC3064 will be available in February. We also entered the military market in 1988, delivering the first MIL-STD-883 Class B spec parts late in the year. We upgraded our software and added new schematic capture packages. We ported software to Sun and Mentor workstations. Over twenty third-party vendors have announced design support for our Programmable Gate Arrays.

We doubled the installed base of Xilinx development systems to more than 3000. These are installed in 1500 companies, and the list of users includes almost every major electronics firm in the world.

For manufacturers of scientific and medical instruments, industrial automation and military projects, Xilinx Programmable Gate Arrays continue to be the clear leader in delivering the advantages of VLSI without the cost and schedule disadvantages of custom ICs. More importantly, Xilinx LCAs are now being used by high volume manufacturers of computer peripherals, office automation, and telecom products as well as PC plug-in boards.

This year we will bring you enhanced software products, more automatic where you want them to be, but also with more powerful user interaction wherever you prefer that.

We are working hard on device speed improvements, modest ones in the immediate future, dramatic ones later on. We are also working on enhanced technologies, on new architectures, higher densities and additional package options.

Xilinx's goal is to continue to be THE Programmable Gate Array company, to grow profitably and to increase its base of satisfied customers.

We expect to bring you many new products in 1989 and hope for your continued support.

Component Availability (February 1989)

		48 PIN		68 PIN		84 PIN		132 PIN	175 PIN	
		PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PLCC	CERAMIC PGA	CERAMIC PGA	CERAMIC PGA	
		-PD 48	-CD 48	-PC 68	-PG 68	-PC 84	-PG 84	-PG 132	-PG 175	Mil-Std-883 Class B
XC2064	-33	C	I M	C I	C I M					
	-50	C	I	C I	C I M					
	-70			C I	C I					
	-100	samples 2Q89: production 3Q89								
XC2018	-33			C I		C I	C I M B			Now
	-50			C I		C I	C I M B			Now
	-70			C I		C I	C I			
	-100			samples 2Q89: production 3Q89						
XC3020	-50			C I		C I	C I M			B: Mar'89
	-70			C I		C I	C I			
	-100			samples 2Q89: production 3Q89						
XC3030	-50					Feb.'89 *	Feb.'89			B: 2Q89
	-70					Feb.'89 *	Feb.'89			
	-100					samples 2Q89: production 3Q89				
XC3042	-50					C I	C I	Feb.'89		B: 2Q89
	-70					C I	C I	Feb.'89		
	-100					samples 2Q89: production 3Q89				
XC3064	-50							Mar.'89		B: 3Q89
	-70							Mar.'89		
	-100							S:2Q89 P:3Q89		
XC3090	-50								C I M	B: 2Q89
	-70								C	
	-100								2Q89/3Q89	

LCA Package and Temperature Options

C	=	Commercial	0°C To 70°C
I	=	Industrial	–40°C To 85°C
M	=	Military	–55°C To 125°C
B	=	Military	MIL-STD-883, class B

XC1736 PD8C (Plastic 8-Pin Mini-DIP, -40°C to +85°C)
XC1736 CD8C (Ceramic 8-Pin Mini-DIP, -55°C to +125°C)

* **Engineering Samples Now**

Shaded Areas: Not Meaningful

Current Software List

DS21 XACT ver. 2.12
XACT ver. 2.12
XC3090ES die files
DOS 16/M Loader ver. 2.49
XC3030/XC3042 die files

DS22 16K-gate P/C-SILOS ver. 2.10
P/C-SILOS ver. 3C.7
XNF/LCA ver. 2.11
XNF2SILO ver. 2.10

DS122 5K-gate P-SILOS ver. 2.01
P-SILOS ver. 2C.5 rev 2
XNF/LCA ver. 2.11
XNF2SILO ver. 2.01

DS23 ADI ver. 2.10
APR ver. 2.13
XNF/lca interface ver. 2.11
Logic Synthesis ver. 1.00

DS23-AP1 ADI ver. 2.1
for Apollo DN3000 and DN4000

DS23-SN1 ADI ver. 2.1
for SUN 2 and SUN 3 workstations

DS28 XACTOR ver. 2.10
XACTOR 2.10

DS31 FutureNet DASH ver. 2.10
FutureNet interface and library
PIN2XNF ver. 2.01

DS32 Schema II+ Interface and
Library ver. 2.13

DS33 Daisy Interface (DNIX) ver. 1.04

DS34 Mentor Interface ver. 1.10

DS35 OrCAD/SDT Interface ver. 1.0
(For SDT ver. 3.1 and higher)

DS40 FutureNet TTL Library ver. 1.0

DS51 ver. 2.13
DS52 ver. 2.11
DS21 XACT ver. 2.12

DS52 ver. 2.11
Schema II+ ver. 2.13
DS32 Schema II Interface ver. 2.13
DS23 ADI ver. 2.10

DS53 ver. 2.12
DS54 ver. 2.13
DS21 XACT ver. 2.12

DS54 ver. 2.13
DASH-LCA ver. 4.02r
DS31 FutureNet Interface ver. 2.01
SD23 ADI ver. 2.10

DS81 Ser. Config PROM Programmer
SCP Programmer ver. 2.00
XPROM ver. 2.00

XC30xx-100 Preliminary Specifications

Expect samples in 2Q89, production quantities in 3Q89

CLB SWITCHING CHARACTERISTIC GUIDELINES

	Description	Speed Grade	Symbol	-50 Min	-50 Max	-70 Min	-70 Max	-100 Min	-100 Max	Units
		1	T_{ILO}		14		9		7	ns
CLB Logic input	Combinatorial	9	T_{RIO}	10	15	7	10	7	7	ns
Reset direct	CLB output Reset Direct width*	13	T_{RPW}		35		25		17	ns
	Master Reset pin to CLB out		T_{MRQ}							ns
CLB K Clock input	To CLB output	8	T_{CKO}		12		8		7	ns
	Additional for Q returning through F or G to CLB out		T_{QLO}		11		7		5	ns
	Logic-input setup	2	T_{ICK}	12		8		7		ns
	Logic-input hold	3	T_{CKI}	1		1		0		ns
	Data In setup	4	T_{DICK}	8		5		4		ns
	Data In hold (1)	5	T_{CKDI}	6		4		2		ns
	Enable Clock setup	6	T_{ECCK}	10		7		5		ns
	Enable Clock hold	7	T_{CKEC}	1		1		0		ns
	Clock (high)*	11	T_{CH}	9		7		5		ns

BUFFER (Internal) SWITCHING CHARACTERISTIC GUIDELINES

	Description	Symbol	-50 Min	-50 Max	-70 Min	-70 Max	-100 Min	-100 Max	Units
				9		6		4	ns
Clock Buffer**	GCLK, ACLK			8		5		4	ns
TBUF**	Data to Output								
	(Long line buffer)			34		22		14	ns
	Three-state to Output			17		11		7	ns
	Single pull-up resistor								
	Pair of pull-up resistors			6		4		3	ns
Bi-directional	BIDI								

IOB SWITCHING CHARACTERISTIC GUIDELINES

	Description	Symbol	-50 Min	-50 Max	-70 Min	-70 Max	-100 Min	-100 Max	Units
Pad (package pin)	To inputs TCLKIN, BCLKIN	3	T_{PIDC}	5		3		2	ns
	To inputs DIRECT IN		T_{PID}	10		7		4	ns
I/O Clock	To I/O RI input (FF)	4	T_{IKRI}	10	20	7	17	6	ns
	I/O pad-input setup	1	T_{PICK}	30	0		0		ns
	I/O pad-input hold	2	T_{IKPI}	0		13		10	ns
	To I/O pad (fast)	7	T_{OKPO}	18	10		9		ns
	I/O pad output setup	5	T_{OOK}	15	0		0		ns
	I/O pad output hold	6	T_{OKO}	0					ns
Output	To pad (enabled fast)	10	T_{OPF}	14		10		6	ns
	To pad (enabled slow)	10	T_{OPS}	33		25		23	ns
Three-state	To pad begin hi-Z (fast)	9	T_{TSHZ}	12		8		8	ns
	To pad valid (fast)	8	T_{TSON}	20		14		12	ns
Master Reset (Package Pin)	To input RI	13	T_{RRI}	45		30		20	ns
	To output (FF)	15	T_{RPO}	55		37		28	ns

-100 INTERCONNECTS ARE APPROXIMATELY 25% FASTER THAN ON -70 PARTS.

30 MHz Binary Counter Uses Less Than One CLB per Bit

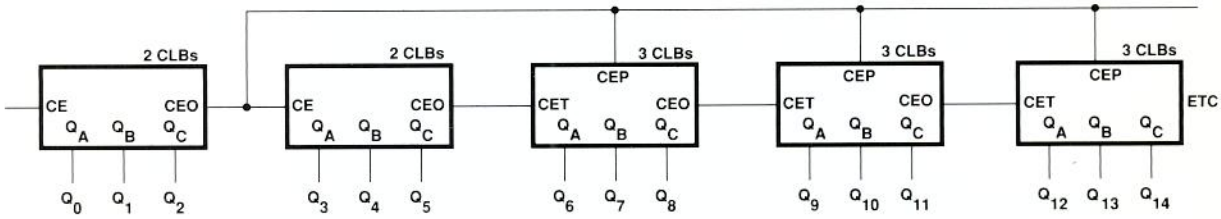
Borrowing the concept of Count Enable Trickle/Count Enable Parallel that was pioneered in the popular 74160 TTL-MSI counter, a fast non-loadable synchronous binary counter of arbitrary length can be implemented efficiently in the XC3000 series CLBs. For best partitioning into CLBs, the counter is segmented into a series of tri-bits.

The least significant (i.e. the fastest changing) tri-bit has a Count Enable Output (CEO) that is routed to all the Count Enable Parallel (CEP) inputs of the whole counter.

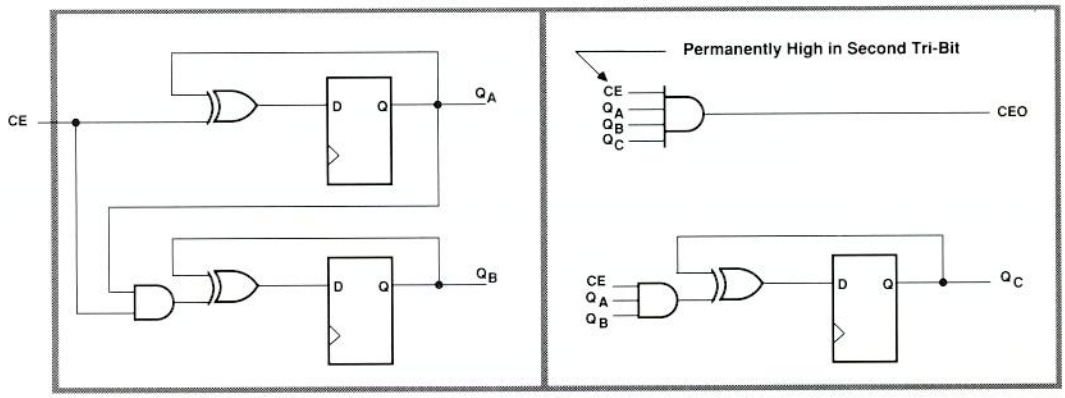
Each Count Enable Output from any other tri-bit drives the Count Enable Trickle (CET) input of its more significant neighbor. The clock causes any tri-bit to increment if all its Count Enable inputs are active (CEP and CET, if both exist). The CEO is active when all three bits are set *and* CET is High. CEP does not affect CEO.

The least significant tri-bit thus stops the remaining counter chain for 7 out of 8 incoming clock pulses, allowing ample time for the CEO-CET ripple-carry chain to stabilize. Max clock rate is determined by the first tri-bit's Clock-to-CEO delay (TCKO + TILO), plus the CEP input set-up time for all other tri-bits (TICK), plus the routing delay of the CEP net. In a –70 this sum can be below 32 ns. The higher tri-bits are not speed critical if they propagate the CET signal in less than eight clock periods, easily achievable through 20 tri-bits, i.e. 60 bits.

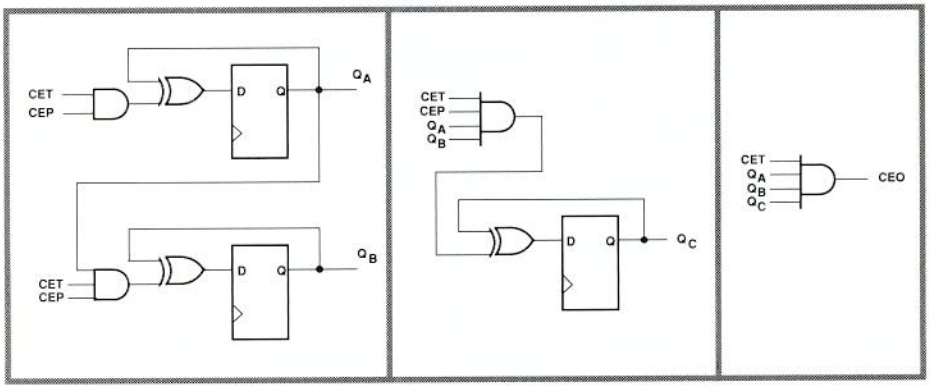
The two least significant tri-bits each have a single Count Enable input; they fit, therefore, in only two CLBs each. The higher tri-bits have two Count Enable inputs (CEP and CET) and require three CLBs. P.A.



30 MHz Non-Loadable Binary Counter, Expandable up to 60 Bits



First and Second Tri-Bits Use Two CLBs Each

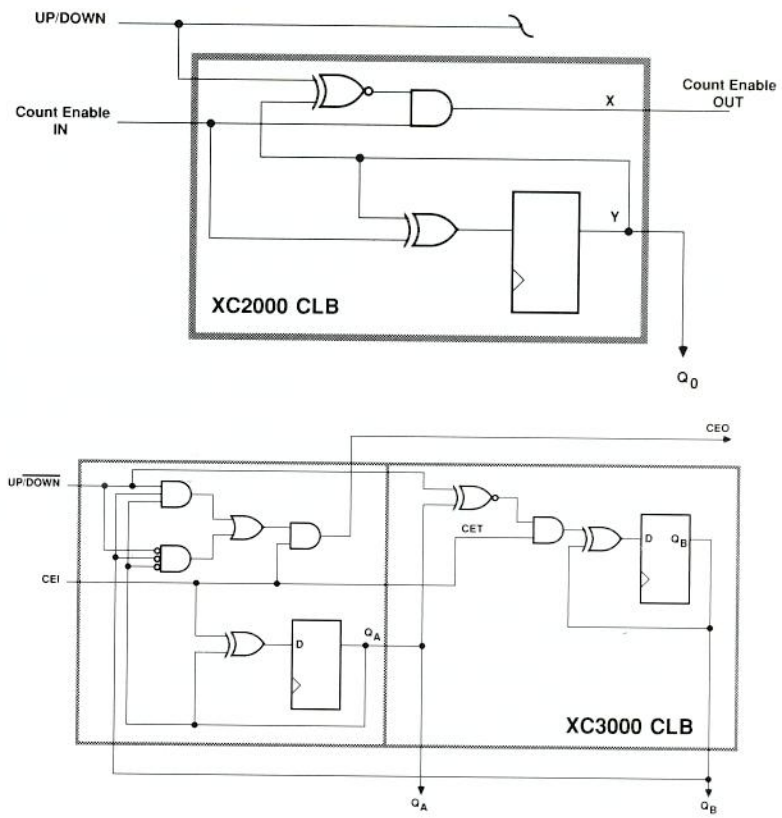


All More Significant Tri-Bits Use Three CLBS

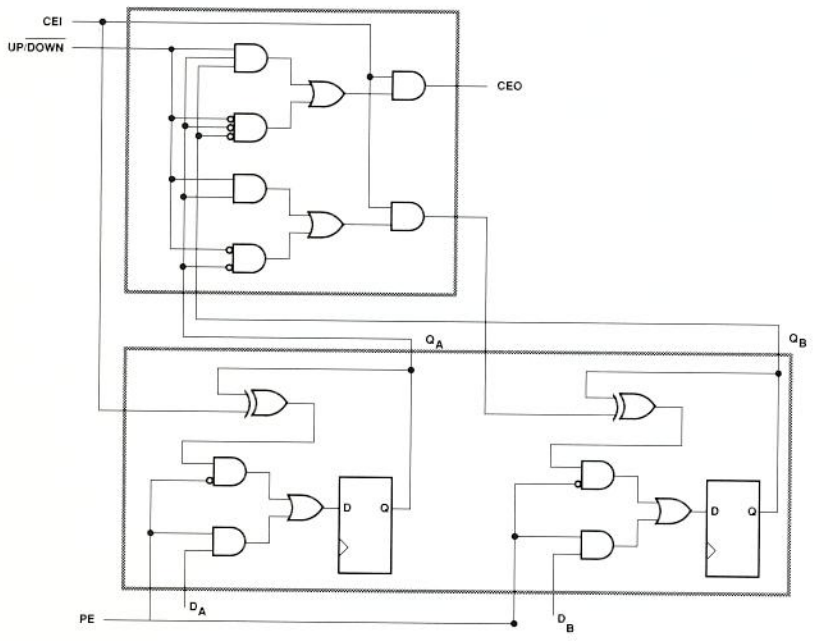
Three Synchronous Up/Down Counter Designs Use Only One CLB per Bit

A fully synchronous resettable but non-loadable up/down counter of arbitrary length can be implemented with only one XC2000 CLB per bit. This design cascades the toggle information from the least significant toward the most significant position. Such an architecture reduces the maximum clock rate for longer counters, from 30 MHz for 2 bits, to 10 MHz for 8 bits, down to 5 MHz for 16 bits, assuming a –70 part. This simple design is, therefore, not suited for high speed clocking, but it generates fully synchronous outputs; i.e. all flip-flops clock on the same edge.

The better functionality of the XC3000 CLBs can cut the cascaded toggle control delay in half by looking at two counter bits in parallel. This doubles the max frequency for a given counter size. A 16-bit counter in a –70 part can count 10 MHz, guaranteed worst case.



Loadable Up/Down Counter



The five-input function generator of the XC3000 family CLBs makes it possible to build expandable fully synchronous *loadable* up/down counters of arbitrary length using only two CLBs per two bits, i.e. one CLB per bit.

The basic concept is similar to the non-loadable up/down counter described above. The function generator driving the counter flip-flop has two additional inputs (Parallel Enable and Data). The cascaded toggle control circuit is moved to a separate CLB which serves two counter bits simultaneously. This cuts the effective ripple delay in half.

The CEP/CET speed enhancement cannot be used on up-down counters that might reverse their direction of count in any position. They can, therefore, not guarantee a defined number of clock periods for the ripple-carry chain to stabilize.

P.A.

Programmable Sine Wave Generator

Sine wave frequency synthesizers are used in many applications, like telecom and navigation. A sine wave of programmable frequency can be generated by sequencing through a look-up table in ROM that drives a digital-to-analog converter (DAC).

The simplest and most flexible arrangement uses an accumulator to access the look-up table. (Remember, an accumulator is an adder/register structure that adds an input value to the register content each time it is clocked.) The desired frequency is presented as a constant (K) to the accumulator input. Changing K results in an instantaneous frequency change (as a result of the next clock edge) but no sudden phase change, no "clicks." This is mandatory in modems.

Here is one design example that fits into 30 CLBs, less than half of an XC3020: The objective is to generate any frequency that is an integer multiple of 1 Hz, the highest frequency being around 250kHz. The sine wave look-up table has 64 entries for a 2π (360°) period, i.e. a resolution of 5° to 6°. It represents the amplitude as a 9 bit binary word (8 bits plus sign). These are reasonable parameters, but each of them could easily be modified by an order of magnitude without changing the design concept. The look-up table consists of a 64 x 8 ROM (really a 16 x 8 ROM plus XORs on the address inputs and data outputs) addressed by the 6 most significant outputs of the accumulator.

The ratio of max frequency to frequency resolution determines the size of the accumulator; in this case it is 250kHz ÷ 1Hz = 250K or 18 bits. That would, however, give only one look-up per period at the top frequency; this design, therefore, adds four bits to the accumulator in order to guarantee sixteen look-ups even at 250 kHz. The accumulator; clock rate is then determined by the frequency resolution (1 Hz) and the accumulator length (22 bits): If the accumulator increments by one for every clock period, it must step through the whole look-up table once per second. The clock frequency is, therefore, 2^{22} Hz = 4.194304 MHz.

The four most significant accumulator bits have no data inputs; they can, therefore, be implemented as a counter. The look-up table stores only the first quadrant (90°) of a sine wave, the other three quadrants are generated by reversing the address sequence (XORing the addresses) and/or reversing the sign of the output (XORing the outputs).

Better frequency resolution can be achieved by adding stages to the LSB end of the accumulator (1 CLB for each doubling of the resolution.) No change in clock frequency.

Higher max frequency can be achieved by adding to the MSB end of the accumulator and doubling the clock frequency for every additional bit.

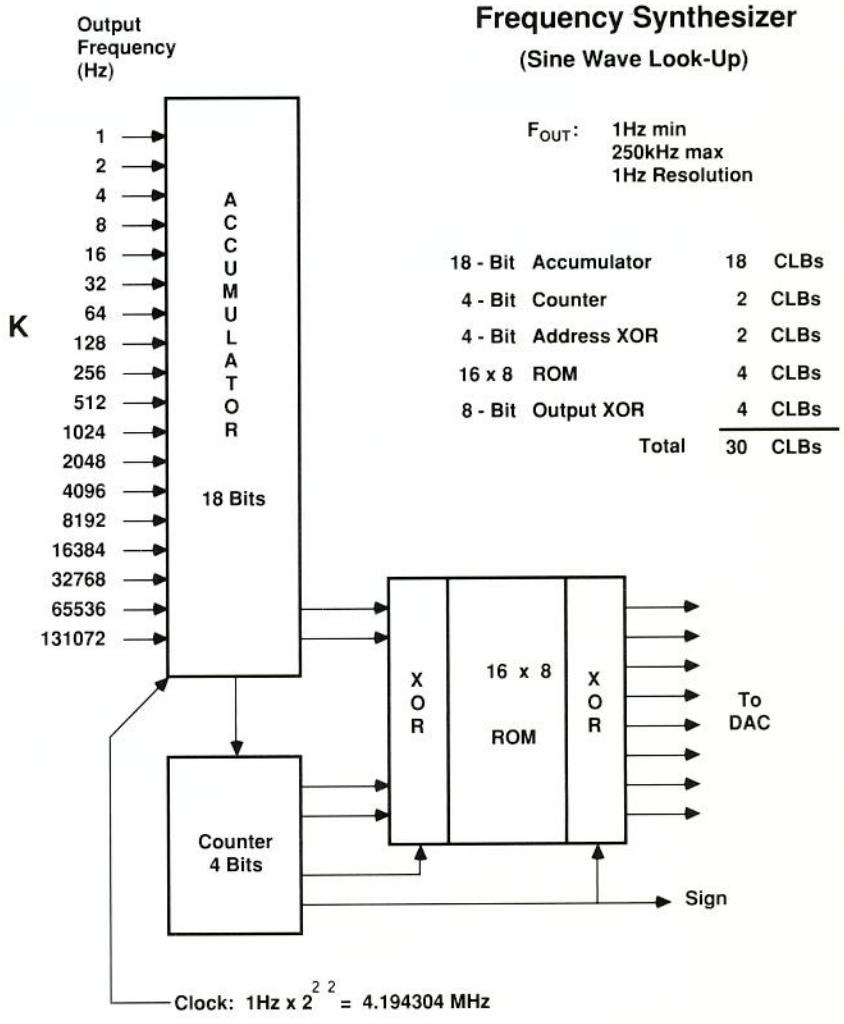
The time-granularity of the look-up table can be doubled to 32 entries per quadrant, increasing the table from 4 CLBs to 8.

The amplitude granularity of the look-up table can be changed in either direction by changing the number of planes in the look-up table.

Obviously, the look-up table can also store other wave shapes and can be reprogrammed dynamically.

These hints should allow any designer to custom tailor a similar frequency synthesizer.

P.A.



Four-Bit Barrel Shifter in Only Four CLBs

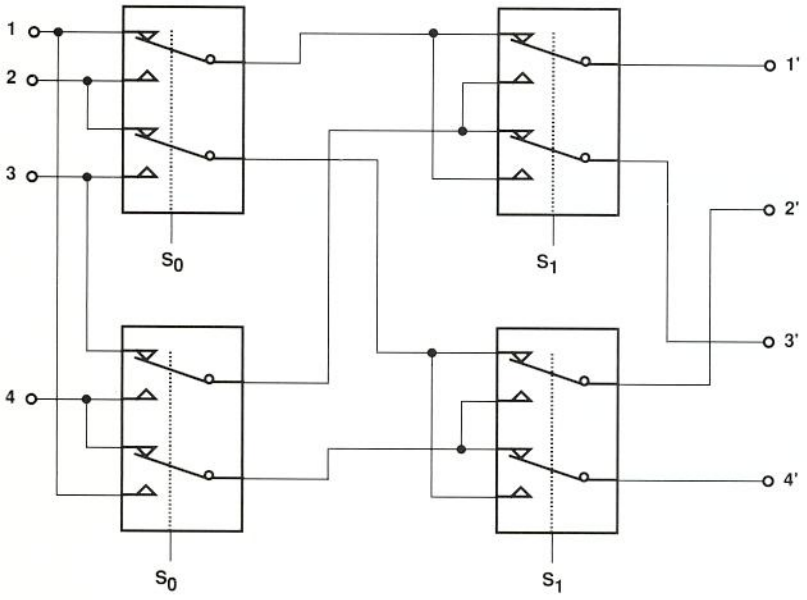
Barrel Shifters are used in a variety of applications; they are also popular benchmarks. A Barrel Shifter is a combinatorial circuit, a glorified multiplexer, that can rotate parallel incoming data by any number of positions.

A four-input barrel shifter has four data inputs, four data outputs and two control inputs that specify rotation by 0, 1, 2, or 3 positions. A brute-force design would use four 4-input multiplexers, since each output can receive data from any input. Each four input multiplexer requires two XC3000 family CLBs, for a total of eight CLBs.

There is, however, a smarter method that reduces the design to only four CLBs. The key to this approach lies in the two signal crossovers at the input and output of the second level CLBs.

Eight-Bit Barrel Shifter in 12 CLBs

The 4-Bit Barrel Shifter design can be extended to eight bits. A first level shifter consisting of four CLBs rotates the eight inputs by one position, controlled by the least significant control input. Two interleaved 4-Bit Barrel Shifters then take the eight outputs from the first level and rotate them by 0, 2, 4 or 6 positions. P.A.

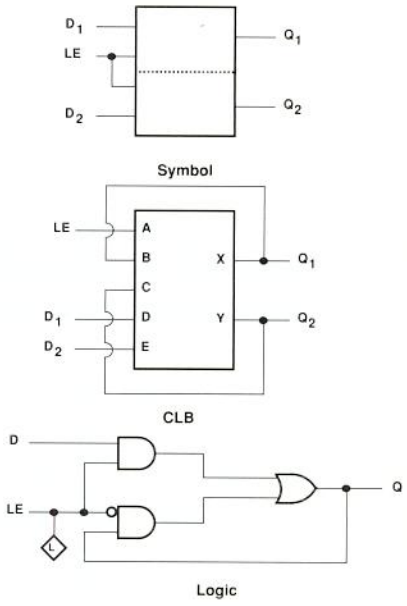


Four-Bit Barrel Shifter in Four CLBs

Two Latches in One CLB

Page 6-25 of the new Data Book shows how to make a latch out of the combinatorial logic in a 3000-family CLB. It fails to mention that you can also get two basic latches from one CLB.

The two latches have a common Latch Enable, but individual data inputs and Q outputs. Any of these inputs and outputs can be active High or active Low. It is even possible to have one of the two latches with an active High Latch Enable, the other one with an active Low. S.K.



F = ~A • B + A • D

G = ~A • C + A • E

Equations

Adder Logic Truth Tables

Pages 6-22/23 of the 1988 Xilinx Data Book describe 4, 8 and 16 bit adders, but do not show detailed information on the logic inside the CLBs.

We have received several calls asking for more details. Here are the truth tables for the 8-bit adder on page 6-22.

After adjusting the suffixes appropriately, the truth table for the three CLBs generating S_2 and S_3 is identical with that for the CLBs generating S_0 and S_1; and the truth table for the bottom three CLBs is identical to that of the three CLBs generating S_4 and S_5.

We should have been more explicit in labeling the carry outputs C_2, C_4, C_6, etc., instead of just CARRY. The suffix should always designate the binary weight of the signal.

The 16-bit adder is a natural extension of the 8-bit adder.

P.A.

OUTPUTS	INPUTS C_0	A_0	B_0	A_1	B_1
$S_0 = 1$	1	0	0	x	x
	0	1	0	x	x
	0	0	1	x	x
	1	1	1	x	x
$S_1 = 1$	x	1	1	0	0
	1	x	1	0	0
	1	1	x	0	0
	x	0	0	1	0
	0	x	0	1	0
	0	0	x	1	0
	x	0	0	0	1
	0	x	0	0	1
	0	0	x	0	1
	x	1	1	1	1
	1	x	1	1	1
	1	1	x	1	1
$C_2 = 1$	x	x	x	1	1
	x	1	1	1	0
	1	x	1	1	0
	1	1	x	1	0
	x	1	1	0	1
	1	x	1	0	1
	1	1	x	0	1

OUTPUTS	INPUTS C_4	A_4	B_4	A_5	B_5
$S_4 = 1$	1	0	0	x	x
	0	1	0	x	x
	0	0	1	x	x
	1	1	1	x	x
$S_5 = 1$	x	1	1	0	0
	1	x	1	0	0
	1	1	x	0	0
	x	0	0	1	0
	0	x	0	1	0
	0	0	x	1	0
	x	0	0	0	1
	0	x	0	0	1
	0	0	x	0	1
	x	1	1	1	1
	1	x	1	1	1
	1	1	x	1	1
$CG_6 = 1$	x	x	x	1	1
	x	1	1	1	0
	x	1	1	0	1
$CP_6 = 1$	x	0	1	1	0
	x	1	0	1	0
	x	0	1	0	1
	x	1	0	0	1

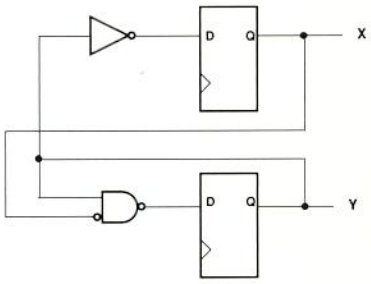
OUTPUTS	INPUTS C_{in}	LOWER CP	LOWER CG	HIGHER CP	HIGHER CG
LOWER CARRY	x	x	1	x	x
	1	1	x	x	x
HIGHER CARRY	x	x	x	x	1
	x	0	1	1	x
	1	1	0	1	x

High Speed ÷ 3 Counter in One CLB

Some microprocessors require a 2/3 duty cycle clock, most conveniently and reliably generated by dividing a three times faster crystal oscillator frequency by three.

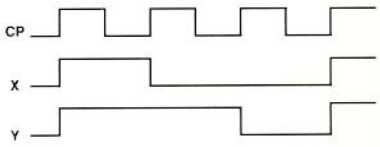
The design described below uses one XC3000 series CLB to generate a 1/3 High duty cycle signal on the X output, and a 2/3 High duty cycle signal on the Y output. This is just one of many possible implementations. Max clock frequency is 60 MHz in a –70 device.

P.A.



Y	X
0	1
1	1
1	0
0	0

LOOP



ADI Update Coming Soon

We have increased the effectiveness of our design tools by creating a new version (version 2.2) of the DS23 Automated Design Implementation (ADI) package. This revision of ADI improves many of the existing features, while adding powerful new capabilities. The result is a better partitioned design and improved automatic placement and routing algorithms. Here are some highlights of the new software:

- Logic partitioning can be directed from the schematic editor. New mapping symbols can be included in the schematic to control the mapping of the design's logic into the configurable logic blocks of the LCA architecture. The new schematic symbols for controlling partitioning are currently available for the Dash-LCA and FutureNet DASH editors. Other schematic interfaces will be updated soon.
- The Xilinx Netlist File (XNF) format now fully supports hierarchy. This provides greater flexibility by allowing designs to be created in distinct sections that can be independently mapped into the LCA, resulting in more efficiently partitioned designs. It will also allow back annotation in future releases, which aids in the development of interfaces to timing simulators.
- The Automatic Place and Route (APR) program has been improved, particularly for large and/or sparse designs. This new version has consistently routed designs 100% that were not fully routed by earlier versions of APR.

The new ADI software is now in limited beta-siting. Production release and customer updates are planned for April.

Unconnected Inputs in a FutureNet May Cause a SILOS Error

If you are performing a unit delay simulation with PC-SILOS and there are unconnected inputs in your design, they will appear in the default .DAT file. If you remove the

```
unknown_signal_name .CLK O S*
```

instruction from the .DAT file, you may see the following error message when simulating the design: level 2:

```
**Input "unknown_signal_name"
for function "signal_name2" **
is not a node name
```

At the beginning of a SILOS simulation, all inputs must be set to a state in order to define the existence of the node. Until it is defined, the node does not exist or "is not a node name."

Leaving pins unconnected in a schematic is common when only part of a Xilinx macro is used. When you run XNF2LCA, the unused logic is eliminated and the signal name will not exist in the simulator. However, if you run PIN2XNF and XNF2SILO to perform unit delay simulation before running XNF2LCA (which is an acceptable procedure), the above problem may occur if unconnected inputs exist.

If you are not sure about unconnected inputs in your design, we recommend that you run the FutureNet DRC (Design Rule Checker) with the -p option. This will report any unconnected pins in the .DRC file. You should run DRC on each module of your design.

If you find that unconnected inputs exist, do not remove the corresponding .CLK statement in the default .DAT file.

Bugs in SILOS 3.7

If any name in your design contains the characters < or (, you may get an "unexpected interrupt D" and the signal name will disappear from the screen and the print-out, even though the signal results will still appear.

A new release in February is expected to fix this bug. In the meantime our bulletin board gives you a cure appropriately called FIXSILOS.EXE.

After you have run XNF2SILO, you can run this program on your .SIM file, and it will convert "<" into "[" and also ">" into "]".

It should be easy to avoid the use of an opening parenthesis in a name.

PC-SILOS 3C.7 prints out wrong characters when GFORMAT=1 is used, causing garbage to be printed out onto an H.P. LaserJet. It will print out correctly onto an EPSON FX85.

This problem will be fixed in the next release of PC-SILOS. A temporary fix is on our Bulletin Board.

Run FIXSILOS.EXE on any SILOS design. OUT graph files to convert the incorrect characters into printable ASCII.

Beware of Software Version Mismatch

Many users fail to install their software updates immediately; they sometimes wait months and install several updates at once. This practice can lead to version mismatch errors. One such mismatch involves XACT 2.12, APR 2.01 and the DOS/16M Loader Update (version 2.49). When XACT 2.12 and the Loader update are installed, APR 2.13 must also be installed. APR 2.01 will not work with these versions of XACT and DOS/16M. When APR 2.01 is invoked, the following (or similar) error messages will result:

```
Partinfo file read error .....
Loader expected version 2.0
found 3.0 ..... Loader execution
terminated.
```

Once APR 2.13 is installed, everything should work fine!

Software Security on Workstations

Unauthorized copying of our PC-based software is discouraged through the use of software protection "keys" that attach to the parallel printer port of the PC. The "keys" contain timing elements that are triggered and periodically sampled by the software in order to verify that the use is legitimate. Since most workstations are part of computer networks, a different protection scheme is required for our workstation-based software.

Workstations, such as the Sun and Apollo computers, are often linked via computer networks, where each workstation on the network is referred to as a "node." To identify specific workstations on a network, a "node name" and "node ID" are assigned to each workstation. The *node name* is chosen by the user and can be changed, analogous to customized auto license plates. The *node ID* is assigned to the workstation by its manufacturer and cannot be changed, analogous to auto engine block identification numbers. Since node IDs are unique, they can be used to enable our software in place of a hardware key.

Software protection of workstation-based programs is implemented through the use of an "authorization code," a string of 7 alphanumeric characters provided to the user by Xilinx. The user enters the authorization code into a special file on the workstation. Application software, such as the APR program, checks this file for a valid authorization code each time the application is invoked.

The authorization codes for Xilinx development system products are generated by encrypting the node ID, software product name, and expiration date. The authorization code is, therefore, node specific, product specific, and expires on the specified date. In other words, the authorizaiton code is good only for running the specified programs on the workstation with the specified node ID for a predetermined time.

To enable us to generate the proper authorization code, all purchase orders for workstations-based products should include the appropriate node ID. After you receive the product, telephone or send a FAX to Chris Tully in the Customer Service Department to receive the authorization code. Chris can be reached at (408) 559-7778, ext. 282, or by FAX at (408) 559-7114. This procedure is explained in the installation instructions that accompany the product. Since most software vendors use a similar type of system, you may already be familiar with this method of software protection.

Once you receive the authorization code from Xilinx, use a text editor to enter the node ID, authorization code, and expiration date in a special file named "authorization_list" on your workstation.

For Apollo workstations:

- Node IDs are 4 or 5 alphanumeric characters.
- The directory path to the authorization_list file is /local_user/xilinx/authorization_list.
- All Apollo computers run our Apollo-based software.

  Typically, you will encounter DN 3000, DN 3500, and DN 4000 family workstations.

For Sun workstations:

- Node IDs are 8 alphanumeric characters.
- The directory path to the authorization_list file is /$XILINX/files/authorization_list.
- Only Sun 3 workstations run the DS23-SN1 software: Sun 4 and Sun 386i computers will not execute our software.

  Design files are transferable between the workstations and PCs.

IOB Options

Pages 2–3 and 2–4 of the 1988 Data Book describe the operation of the XC3000 IOBs and their configuration options. This description is not complete: The activation of the passive pull-up is really coupled with the Three-State Enable, giving the following four choices:

- Passive pull-up activated, output buffer permanently three-stated (pin is input only, with pull-up).
- Passive pull-up de-activated, output buffer permanently three-stated (pin is input only, no pull-up).
- Passive pull-up de-activated, output buffer active, i.e. three-state control permanently de-activated (pin is ouput only).
- Passive pull-up de-activated, output buffer controlled by three-state control signal (pin can be I/O).

In other words:

The passive pull-up can only be used on pure inputs, not on I/O pins.

The three-state control logic can be permanently disabled, resulting in a permanently active output.

The other four options (OUT INVERT, THREE-STATE INVERT, OUTPUT SELECT and SLEW RATE) are not interdependent, they operate as described.

5 1/4 or 3/12 Inch? You Get What You Ask For.

Xilinx will create both types of diskettes, 5 1/4 inch as well as 3 1/2 inch, for any new software release.

If you want to receive 3 1/2" diskettes, please indicate so on your registration card.

The Tie Option Demystified

Before generating the configuration bit stream, the user has the option to tie or not tie the design. To "tie" means to create additional interconnects that terminate all floating transistor inputs or metal interconnects to well-defined levels or signals.

In a tied design all inputs and interconnects are always High or Low, or are connected to a signal that switches between defined levels. In a non-tied design the unused inputs and pieces of interconnect (there usually are more unused ones than used ones) are left floating. This poses no first order problem, since these inputs are really not used. In this respect it differs from the well-known problem of floating TTL inputs that are supposed to generate a High level.

The problem with undefined input levels in CMOS logic is that they may drift to the midpoint between Vcc and ground, half turning on both the pull-down and pull-up transistor, making a CMOS gate draw measurable Icc. Also, such undefined inputs may be affected by crosstalk from adjacent lines, thus increasing dynamic power consumption. An untied design is likely to have increased dc and ac power consumption and increased on-chip noise. That's a good enough reason to spend the extra effort to tie every design.

Unfortunately, there can be problems when XACT tries to tie a design. The software connects any unused input or net either to ground (by using the output from an appropriately configured unused CLB) or, when that is not possible, to a signal that has been declared non-critical. XACT will not tie any additional interconnect to a net marked critical. The XACT FLAGNET command is used to mark "critical" nets which are delay sensitive. XACT assumes that the small speed degradation caused by additional load is not acceptable on any critical net.

When XACT finds it impossible to tie certain pins, it states this in an error message. The user **cannot** ignore this message and generate the bit-stream for a partially tied design.

There are only two choices:

1. Leave the design completely untied or,
2. Modify the design with XACT, so that it can be tied automatically. We do not recommend the first choice.

The better way is to use the interactive routing capability of XACT to assist the tie-process. The error message gives very specific information about the—usually few—untied pips in the design. Usually there are fairly easy ways to make the design tie-able: re-route a blocking net, or re-evaluate the critical nature of the nets close to the untied pips. They may have been labeled "critical," but they may actually tolerate the small additional delay caused by the connection of additional routing to an otherwise unconnectable input. XACT is an excellent tool to make these routing changes and evaluate their impact on device performance.

With some patience and creativity, any design can be tied.

It might then be a problem to get the manually modified design to pass the Design Rule Check (DRC) program, which has to occur before tie can be invoked again and finally, a bit-stream can be generated.

There is an inconsistency in our software: DRC, which occurs normally before Tie, uses a very pedantic interpretation of the design rules, easily violated by Tie.

For reasons too obscure to explain here, DRC treats any connection to an *unused* CLB input as a "fatal error," even though a tied design must connect all CLB inputs, used or unused. If the manual modification mentioned above involves connecting an unused CLB input, the design will fail DRC and therefore, cannot generate a bit-stream.

Until the software is modified later this year, the user has only one way out of this dilemma: Incorporate the offensive input into the logic equation of the CLB, by creating redundant terms. F = (C + D) * (A + ~A) for example, will cause DRC to think that A is used, even though it has no effect on the operation of the circuit. Similar tricks may be used on IOB input pins. Not elegant, but straightforward.

There is one additional obscure problem that has nothing to do with the device, but is created by the IBM-PC: In a very sparsely populated XC3090 (only in this device) the tie option can create a net so big that it causes the message:

```
stackstop: XACT operation may
fail or yield incomplete results
```

to be displayed. This message is displayed when XACT detects that it is about to overflow the 64K byte run-time stack. This is a result of the infamous 64K segment size limitation inherent to Intel microprocessors. A '386 machine is no help here since our software presently runs the '386 CPU in '286 mode.

There are two solutions:

Before invoking the tie-option, you might create some large nets that tie all unused long lines and many other lines either High or Low or to any uncritical net. This might eliminate the automatic tie stack overflow.

The DRC, MAKEBITS, and TIE functions are in the process of being ported to the Sun 3 series workstations. These are built around the Motorola 68020, which has a linear address space. The stack problem will not exist on this machine. Schedules for the release of this standalone, non-graphics DRC/MAKEBITS/TIE program are being developed.

I/O Clocking:

Don't Let the Software Confuse You

In all XC3000 family devices each IOB has a choice of two clock sources. These two clock signals are common for each edge of the chip; the same source can of course also be connected to several edges. The clocking polarity of any of these two lines can be selected by configuration, but only for the whole edge of the chip, not for each individual IOB. The XACT user interface incorrectly implies that the designer has the option of selecting clock polarity for each IOB. Not true.

The user can configure each individual IOB storage element as either flip-flop or latch, but the clocking polarities of these two options are interrelated:

If a clock line acts as a rising edge clock for a flip-flop, connecting it to a latch makes the latch transparent while the clock is LOW, latched while the clock is HIGH.

Similarly, a falling edge flip-flop clock becomes an active High Latch Enable. *This relationship is not obvious*, in fact it is surprising, but it optimizes performance. (The latch is really the master portion of the master-slave flip-flop.)

XACT does not make this relationship clear: If flip-flops and latches are to be driven from one clock line, they must be specified with opposite clock polarities: IK for rising edge triggered flip-flop, IKNOT for Low transparent latch, or vice versa. Ignoring this restriction will lead to a fatal DRC error.

Conclusion

If the user needs only one clocking signal per chip edge, there is complete freedom to supply any IOB storage element (flip-flop or latch) with either clock polarity. If the user needs two different clocking signals on any chip edge, then each of them is restricted to one (any one) clocking polarity. Flip-flop and latch options are interrelated in their polarity choice.

Three-State vs. Output Enable

The control input that causes an IOB output or Long Line driver to go into the high impedance state is called (active High) "Three-State" in Xilinx literature and in XACT. The same signal is commonly known as (active Low) Output Enable or $\overline{OE}$.

These two signals are identical, i.e. T=$\overline{OE}$, as explicitly stated on page 2-12 of our Data Book.

To put it more bluntly: T is not an active High Output Enable, rather it is identical with an active Low $\overline{OE}$.

"Tristate" is a registered trademark of National Semiconductor who pioneered this concept on TTL outputs in the late sixties. The names "Three-state" or "3-State" are ways around this trademark. The name refers to the third state of an output, beyond active High and active Low.

Unused Pins

Xilinx Programmable Gate Arrays come with an abundance of user I/O pins, from 58 on the XC2064 to 144 on the XC3090. Many applications leave a few, or even many, of these pins unused, but even unused pins need some attention.

Modern CMOS devices have extremely low input leakage current, perhaps only a few nanoamps. (The 10 µA guaranteed specification represents a testing limitation, not a real input current.)

Left disconnected, such an input could therefore float to any voltage. Clamp diodes prevent excursions above the supply voltage and below ground, thus protecting the input gate from destructive breakdown voltages. This leaves the problem of inputs floating uncontrolled between Vcc and ground.

An input voltage close to the threshold value (1.2V for TTL level-compatibility, 2.5V for CMOS level-compatibility) will turn the input buffer partially on, thus creating a static current path from Vcc to ground and causing static power dissipation. Such a biased buffer also acts as a fairly high gain amplifier, making the circuit very susceptible to noise, crosstalk, ground-bounce and other undesirable disturbances.

It is, therefore, advisable to force unused inputs to a proper logic level.

XC2064 and XC2018

1. Leave unconfigured; externally connect to a High or Low level, or
2. Configure as active output driven by an internally defined signal.

XC3000 Family

Same as above, or

3. Configure as input with internal passive pull-up.

Putting unused I/O to use

An unused XC3000 series IOB can be used as part of the on-chip logic, e.g. as a shift register. Note that the associated package pin must be left free, and that the speed is not as high as it is with internal flip-flops.

Multiple I/O pins can also be used to perform the "wired AND" function in conjunction with an external pull-up resistor.

Don't Overshoot or Undershoot

Our Data Book (pages 2-37, 2-82, 2-101, 2-105, 2-118) explicitly forbids input voltage excursions more than 0.5 V outside the supply voltages (below ground, above Vcc). Hardly anybody would try to violate this with a static voltage or current, but many designs show PC board reflections that sometimes exceed these rather tight limits. A better explanation of the problem is therefore in order:

All CMOS I/O pins are clamped against Vcc and against ground through diodes formed by the respective output transistors. Pure inputs have equivalent protection diodes. These diodes prevent any excessive voltage on the gate of the associated input transistor. Without such protection the input gate might accidentally get charged to a voltage that can rupture the gate oxide and thus destroy the input transistor. All modern MOS devices have such input protection.

What happens when the input voltage exceeds the specified limits?

Below -0.5 V the ground clamp diode will start conducting, above Vcc + 0.5 V the Vcc clamp diode will start conducting. These diodes are fairly big and will clamp hundreds of milliamps with a voltage drop of less than 2 V. The problem is that this clamp current can stray into an area of the circuit where it might upset the internal logic. There is no hard data to quantify this concern, but our circuit designers feel uncomfortable about undefined currents of prolonged duration in parts of the circuit that were not designed for it.

Very high clamp currents (more than 100 mA at elevated temperature, more than 300 mA at room temperature) lasting for milliseconds can cause the parasitic bipolar input transistors to be triggered like an SCR, then conducting unlimited Icc and thus destroying the device. Xilinx devices are extremely resistant to this latch-up.

Conclusion:

Try to limit overshoot and undershoot to 0.5 V, the data sheet limit. If these values are exceeded, the clamp diodes will protect the inputs and limit the voltage swing. Large clamp currents of millisecond duration must be avoided at all costs, e.g. by adding current limiting series resistors.

Never drive inputs with active levels above Vcc, even when the Vcc supply is turned off. Strange things might happen during turn-on.

Function Generator Avoids Glitches

The combinatorial logic in all CLBs is implemented as a function generator in the form of a multiplexer, built out of transfer gates. The logic inputs are the select inputs to this multiplexer, while the configuration bits drive the data inputs to the multiplexer.

The Xilinx circuit designers were very careful to achieve a balanced design with similar (almost equal) propagation delays from the various select inputs to the data output.

The delay from the data inputs to the output is, of course, immaterial, since the data inputs do not change dynamically. They are only affected by configuration.

This balanced design minimizes the duration of possible decoding glitches when more than one select input changes. Note that there can never be a decoding glitch when only one select input changes. *Even a non-overlapping decoder cannot generate a glitch problem,* since the node capacitance will retain the previous logic level until the new transfer gate is activated about a nanosecond later.

When more than one input changes "simultaneously," the user should analyze the logic output for any possible intermediate code. If any such code permutation produces a different result, the user must assume that such a glitch might occur and must make the system design immune to it. The glitch might be only a few ns long, but that is long enough to upset an asynchronous design.

If none of the possible address sequences produces a different result, the user can be sure that there will be no glitch.

The designer of synchronous systems generally doesn't worry about such glitches, since synchronous designs are fundamentally immune to glitches on all signals except clocks or direct set/reset inputs.

Die Organization

Since XACT shows our devices in one specific orientation, and we, therefore, refer to the top, bottom, left and right side of the die, here is a list of the number of horizontal rows and vertical columns for all of our devices.

CLB Organization

	Rows	x	Columns
XC2064	8	x	8
XC2018	10	x	10
XC3020	8	x	8
XC3030	10	x	10
XC3042	12	x	12
XC3064	16	x	14
XC3090	20	x	16

Internal Bus Contention

The XC3000 family has internal 3-state bus drivers (TBUFs). As in any other bus design, such bus drivers must be enabled carefully in order to avoid, or at least minimize, bus contention. (Bus contention means that one driver tries to drive the bus High while a second driver tries to drive it Low).

Since the potential overlap of the enable signals is lay-out dependent, bus contention is the responsibility of the LCA user. We can only supply the following information:

While two internal buffers drive conflicting data, they create a current path of typically 6 mA. This current is tolerable, but should not last indefinitely, since it exceeds our (conservative) current density rules. A continuous contention could, after thousands of hours, lead to metal migration problems.

In a typical system, 10 ns of internal bus contention at 5MHz would just result in a slight increase in Icc: 16 bits x 6 mA x 10 ns x 5 MHz x 50% probability = 2.5 mA.

There is a special use of the 3-state control input: When it is directly driven by the same signal that drives the data input of the buffer (i.e. when D and T are effectively tied together as shown on page 2-12 of the new 88 Data Book), the 3-state buffer becomes an "open-collector" driver. Multiple drivers of this type can be used to implement the "wired-AND" function, using resistive pull-up.

In this situation there cannot be any contention, since the 3-state control input is designed to be slow in activating and fast in de-activating the driver.

The timing calculator actually shows a delay to the D input 2 ns shorter than the delay to the T input. This is no real problem, just a case of misleading modeling. The delay to D does not include the buffer delay whereas the delay to T does.

Connecting D to ground is an obvious alternative, but may be more difficult to route.

Please excuse the reversal of the descriptions under figures 15a and 15b on page 2-12 of the Data Book.

Ground Bounce

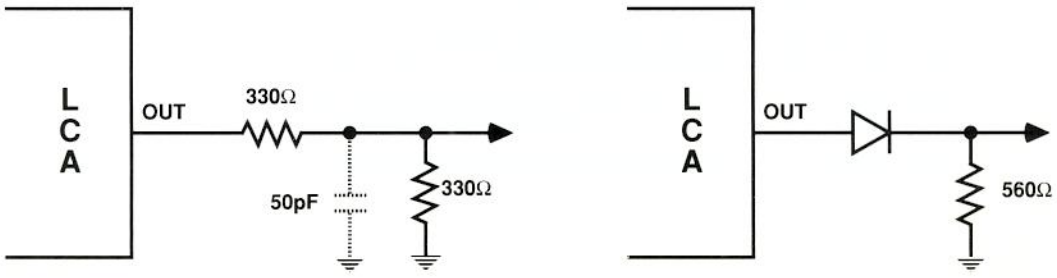
Activating or changing a large number of output pins simultaneously can lead to voltage spikes on the ground and Vcc levels inside the chip.

The output current causes a voltage drop in the supply distribution metalization on the chip, in the bonding wires and the lead frame. Worse is the inductive voltage drop caused by the current change over the bonding wire inductance.

This is a well-known problem not only with fast bipolar or CMOS interface devices, but also with high pin-count gate arrays. It is commonly referred to as "ground bounce", because the change in ground potential is more critical than the equivalent change in Vcc potential. (TTL-oriented systems have far less noise immunity at the Low level than at the High level).

Xilinx circuit designers have given the LCAs a very good Vcc and ground distribution metal grid on the chip, as well as double bonding to every supply pin. Packages below 100 pins have two Vcc and ground pin pairs, packages above 100 pins have eight Vcc and ground pin pairs to reduce supply lead resistance and inductance. What can the user do to minimize ground bounce?

- Provide solid Vcc and ground levels. Use multi-layer boards and good decoupling. *Wire-wrapping the supply connections is an invitation to disaster.*
- Absolutely always connect all Vcc and ground pins.
- Configure outputs XC3000 slew-limited whenever the required performance allows this. This is the default option. Slew-limited outputs reduce transient amplitude by 75%.
- Use CMOS input levels whenever possible. This increases input noise immunity from less than 1 V to over 2 V.
- Stagger the activation or the change of output drivers by deliberately introduced unequal routing delays.
- Move trouble-causing outputs close to a package ground pin in order to minimize the device-internal voltage drop. Move sensitive inputs, like clocks, close to another package ground pin.
- Finally, if there still is a ground bounce problem on a few outputs, attenuate and/or filter these outputs. A 50% attenuater (330Ω, 330Ω) perhaps combined with a 50 pF decoupling of the center point will reduce V_{OL} and calm it down. Changing the upper resistor to a diode might improve the situation even more.



Powerdown Operation

A Low level on the PWRDWN input, while Vcc remains higher than 3 V, stops all internal activity, thus reducing Icc to a very low level:

- All internal pull-ups (on long lines as well as on the I/O pads) are turned off.
- All package outputs are three-stated.
- All package inputs ignore the actual input level and present a 1 (High) to the internal logic.
- All internal flip-flops or latches are permanently reset.
- The internal configuration is retained.
- When PWRDWN is returned High, after Vcc is at its nominal value, the device returns to operation with the same sequence of buffer enable and D/P as at the completion of configuration.

Things to remember

Powerdown retains the configuration, but loses all data stored in the device. Powerdown three-states all outputs and ignores all inputs. No clock signal will be recognized. Any input level between ground and the actual Vcc is allowed.

All internal flip-flops and latches are permanently reset and all inputs are interpreted as High, but the internal combinatorial logic is fully functional.

Things to watch out for

Make sure that the combination of all inputs High and all internal Qs Low in your design will not generate internal oscillations or create permanent bus contention by activating bus drivers with conflicting data onto the same long line.

These two situations are far-fetched, but they are possible and will result in increased power consumption. It is quite easy to simulate these conditions since all inputs are stable and the internal logic is entirely combinatorial, unless latches have been made out of function generators.

Make sure that no applied signal tries to pull any input more positive than the actual supply voltage (Vcc). This would feed Vcc through the input protection clamp diode.

Input Current Is Zero

Some designers keep asking about input current. Let us state bluntly:

The input current is negligible, just nanoamps, if

- the output sharing the same pin is 3-stated,
- the internal pull-up option is not activated,
- the device is not in configuration mode where many pins have internal pull-ups as shown on pages 2-32, 2-80, and 2-81 of our '88 Data Book,
- Vcc is above 4V.

If you ever observe our inputs hogging the drive voltage, you must have done something wrong. Make sure you counted the pin number right—and in the right direction, that you configured the device properly, and that Vcc is up. Then use an oscilloscope and multimeter, but please don't use the phone. Our inputs don't draw any current worth talking about.

Configuring LCAs in Parallel

In the special case where several LCAs contain identical configuration data, they can be configured simultaneously. This reduces configuration time and program size. When the program is stored in an XC1736, just make one LCA the Master, all others the Slave, interconnect all CCLKs, and drive all DINs in parallel from the XC1736.

There are no timing problems, even at 1.5 MHz. Between the 666 ns cycle time and the 400 ns access plus 60 ns set-up time, there are over 200 ns available for additional delay. This accommodates at least 250 pF of additional capacitive loading. The 1 MHz max specification on page 2-48 only applies to READBACK; during configuration CCLK can be up to 1.5 MHz.

Don't Pre-Assign Package Pins

In theory, LCAs offer the system designer the option to pre-assign the package pins and lay out the PC board before completing the detailed design of the LCA.

This method works well when the LCA is sparsely populated and, therefore, has the additional routing resources to accommodate an imposed pin-out.

For typical designs this method does not work well. We have seen many cases where the LCA could not be routed with the imposed pin-out, but could be routed once the pin-out was left free. This leads to daughterboard unscramblers or to a re-layout of the PC board, headaches and expenses that the user would like to avoid.

So, as a rule, wait with the pin-out assignment until the LCA has been routed. The exception to this rule are very sparsely populated designs or designs with very limited I/O.

CCLK Max Low Time

Most of the circuitry in our LCAs is static, i.e. the chip will work down to zero clock frequency.

CCLK is the exception. Its circuitry is half-static, half-dynamic and does not tolerate a Low time in excess of 5 microseconds. For very low speed operation, you can stretch the CCLK High time to any desired value, but keep the Low time short.

LCAs Make Headlines

Application Specific Integrated Circuits, Programmable Logic Devices and, specifically, Logic Cell Arrays by Xilinx have become popular topics in the technical press.

The most extensive description of our LCAs was the cover story of *IEEE Spectrum*, December '88.

Electronic Design of Jan. 12, 1988, presented a survey article on digital system design and mentioned Xilinx LCAs prominently. The opening photograph shows a Datacube board with nine LCAs on it.

New Utilities on the Xilinx Bulletin Board

Several new utilities have been added to the Xilinx bulletin board. These include the capability to download daisy chain configuration streams through the XACT download cable without entering XACT, and the capability to create broadside bitstreams for loading multiple Programmable Gate Arrays in a Slave Mode broadside fashion.

To access these new utilities, you must be a Xilinx registered user and have established an account on the bulletin board. The programs can be found by selecting the **F**)ile command from the main menu and then using the **A**)rea command to select the Utilities file area. The programs are presented in an archived format, so you must have also downloaded PK36.EXE or have a copy of the PKARC program.

Disclaimer

Please note that these utilities have not been passed through the standard Xilinx software QA procedures. Use these utilities at your own risk.

SPRING '89 Seminar Series

Xilinx and Hamilton/Avnet will be sponsoring a series of introductory seminars in April. These half-day seminars will emphasize the XC3000 family architecture and the latest development system products. If you have not yet designed with the XC3000 family, we recommend that you attend. If you already are familiar with the XC3000 family, please recommend this seminar to a friend!

Agenda

- Introduction to Programmable Gate Arrays
- Technical Description (Architecture)
- Performance (Speed, Density, Cost)
- Design Methodology/ Development Systems (including a demonstration)
- Example Applications
- Summary

Schedule

Northeast

Toronto, Ont.	April 10
Montreal, Que.	April 11
Boston, MA	April 12, 13
New Jersey	April 14

Mid-Atlantic

Philadelphia, PA	April 10
Baltimore, MD	April11

Southeast

Atlanta, GA	April 12
Orlando, FL	April 13

Central

Indianapolis, IN	April 10
Minneapolis, MN	April 11
Dallas, TX	April 14

Northwest

Berkeley, CA	April 11
Santa Clara, CA	April 12
Denver, CO	April 12
Portland, OR	April 14
Seattle, WA	April 13

Southwest

Los Angeles, CA	April 10
Orange County, CA	April 14
Phoenix, AZ	April 13

Power Tools for the PC

For many of our customers the XILINX development software is the most sophisticated and demanding software running on their PCs. It may push some users beyond the limits of their understanding of the PC and the details of the MS-DOS operating system (such as AUTOEXEC files and PATH commands).

Here is a good book that will take the mystery out of the PC:

PC Magazine DOS Power Tools
A Bantam Book,June 88
ISBN 0-553-34526-5.

This 1275 page book is not only a good tutorial (200 pages), it describes the DOS commands on 250 pages, adds almost 400 pages of "Power User Secrets" and includes 200 "Utilities DOS Forgot," that are also included on a floppy.

It's a hefty book at a hefty $39.95 price, but may be a bargain when you need it.