

March 1995

## Application Note

## Introduction

The Xilinx XC7336 and XC7318 are high speed EPLDs capable of solving a wide range of logic and state machine problems for today's high-performance digital systems. This application note presents an explanation of the parts, a series of useful design examples and practical details for designing successfully with these parts.

## **XC7336/XC7318 Architecture**

The XC7336 includes 36 macrocells arranged in four blocks and supports pin-to-pin speeds as fast as 5 nanoseconds and clock rates up to 167 MHz. I/O signals can interface with 5 V, 3.3 V or both levels. Current software support includes XEPLD, ABEL and CUPL and additional 3rd party schematic and simulation environments.

Figure 1 shows the global architecture of the Xilinx XC7336. Note the regular structure of four similar high speed function blocks, centrally connected by the Universal Interconnect Matrix (UIM™) and surrounded by pins.

Signals enter and exit on the pins, form logic operations within the function blocks and form connections and logic operations within the UIM. Each section will be briefly discussed, to show key functionality.

## The Interconnect Structure

There are at least three hierarchical sets of signal paths:

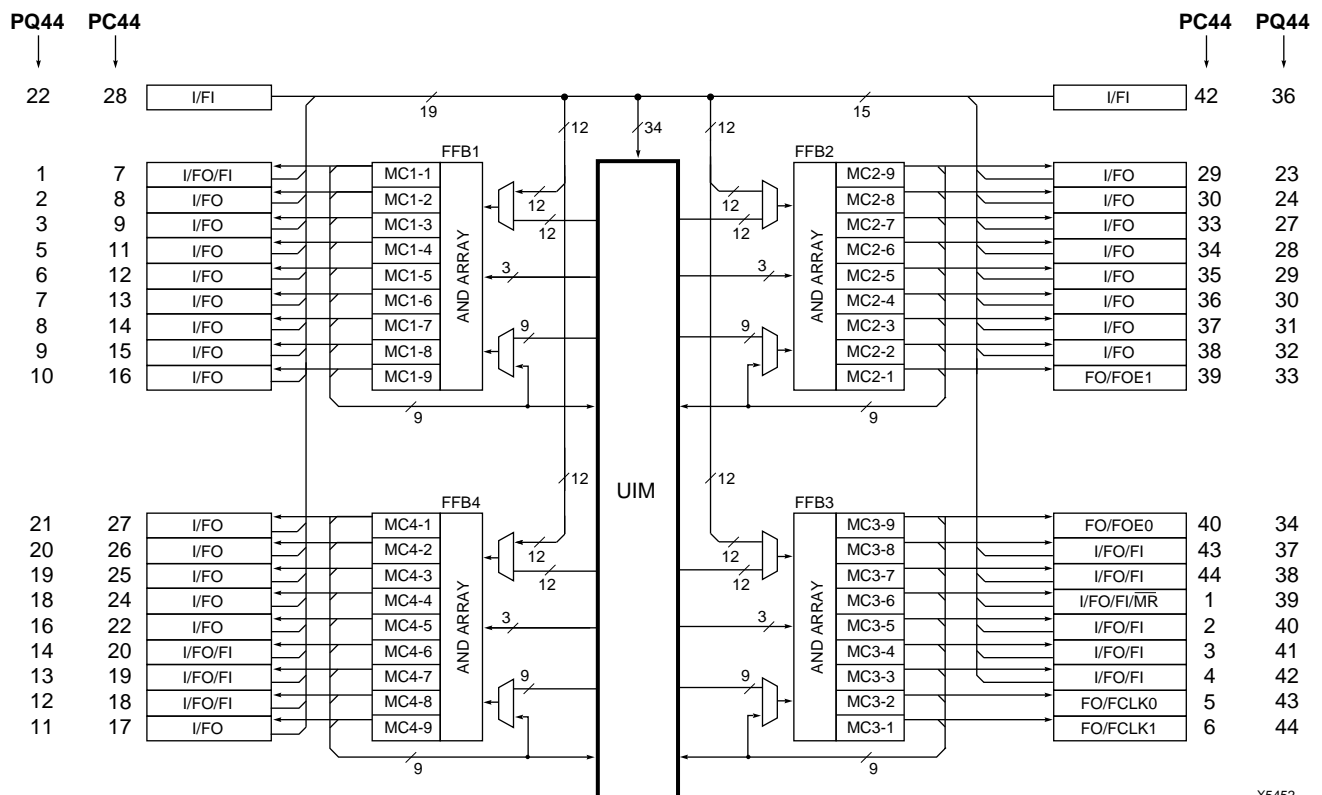
Local - within the function blocks

Global - the UIM

## Global - Fast inputs

## Interconnect Within Function Blocks

Function blocks have 24 input sites. The blocks receive signals from the UIM, local macrocell feedback, block input pins and 12 global fast input pins. Most signals are multiplexed before block entry. The logic blocks themselves can generate 9 signals per function block from the 9 macrocells within each block. Each macrocell signal can drive its own dedicated I/O pin and/or feedback locally within the function block as well as globally by entering the UIM.



### Figure 1. XC7336 Architecture

X5452

### The UIM

The UIM is a crosspoint switch. Every signal entering it can connect to any line exiting it. The UIM has 24 exit points per function block, giving 96 exit points for the XC7336. Any entering signal can be connected to any of the outgoing lines, giving no connection restrictions within the UIM.

### The Fast Inputs

Distributed around the XC7336 is a group of 12 fast input lines. Each line is sourced from a separate pin and directly attached to multiplexers driving into the function blocks. By assigning signals to these input pins, the fastest possible propagation delays occur across the function blocks. Twelve fast input pins covers the largest number of applications requiring fast inputs, and is ideal for fast address decode needs of today's high speed microprocessor systems.

### The Logic Block

The function blocks are groups of 9 macrocells. The 9 macrocells share common input points from the UIM, groups of 9 pins and neighboring product term multiplexers. Each macrocell (Figure 2) within the block is assigned 5 product terms that can be used in a number of ways. The macrocell outputs can then drive output pins

and/or feedback to both the UIM and the function block in which it resides.

### The Macrocell

In the default mode, there are 4 product terms that OR together driving the D input to the macrocell flip flop. This configuration permits the fifth product term to be assigned either to the asynchronous set or reset of the flip flop. This first configuration is the most common one.

Another configuration is to export the 4 product-term cluster to a neighbor, increasing the neighbor's available product terms by 4. Product term exporting is shown in Figure 3. Across the block of 9 macrocells, it is possible to assign all product terms to a single macrocell, taking its total to 36 product terms.

When exporting is used, the remaining product term is reassigned to the D input of the exporting flip flop passing through a series EX-OR gate. By driving one leg of the EX-OR gate with a logical one, the EX-OR inverts making a "NAND" gate at the D input. This remaining logic permits the exporting macrocell to form sums of products when driven from UIM-generated logic.

XC7336 macrocells can also be transformed into T flip flops. The software automatically configures the Q to be

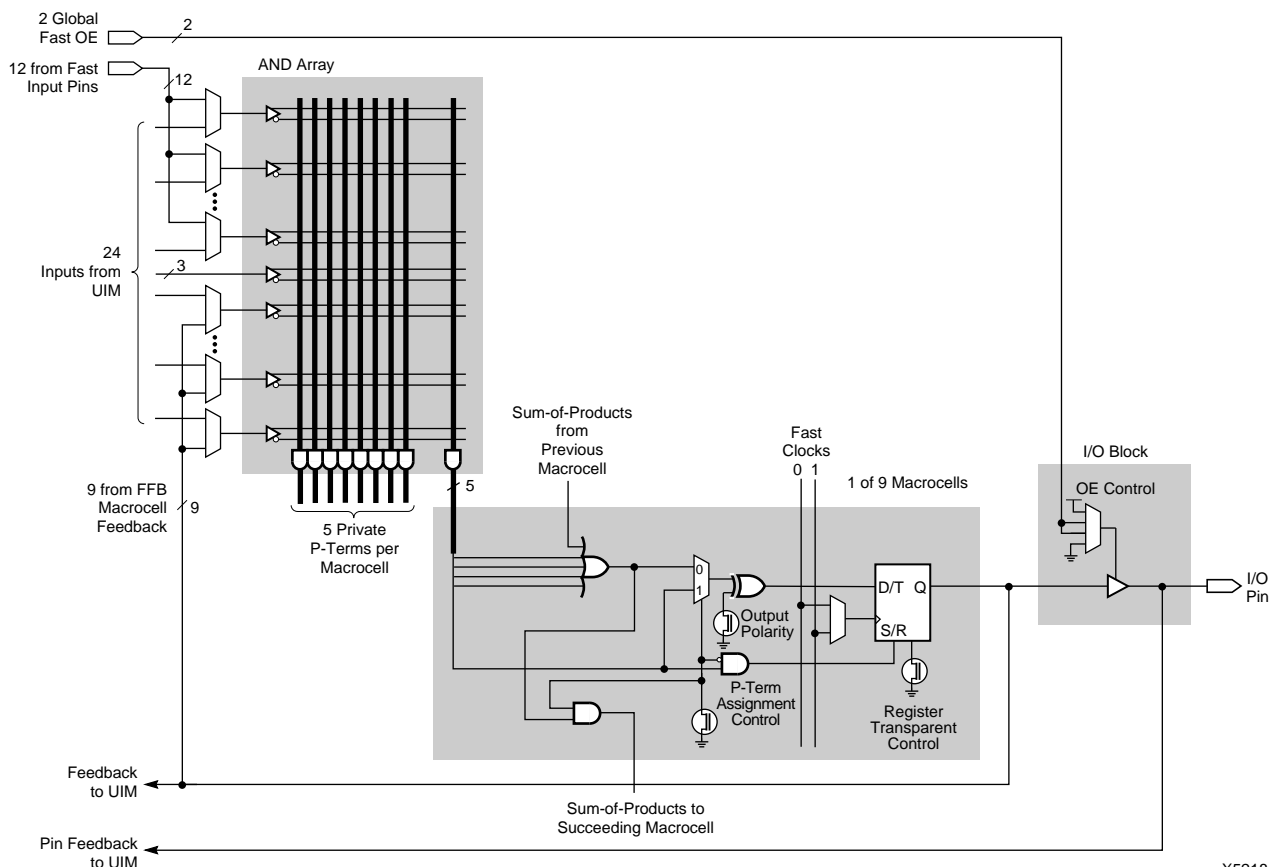
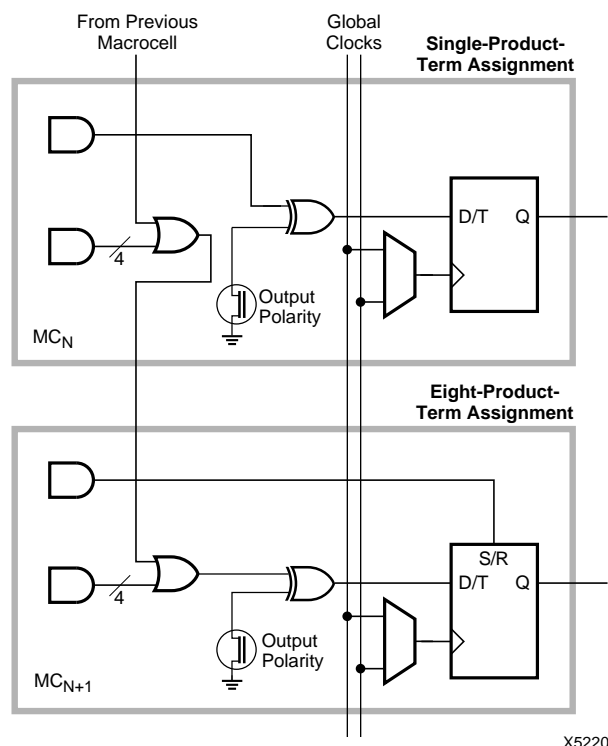


Figure 2. XC7336/XC7318 Macrocell Architecture

X5218



**Figure 3. Product Term Exporting**

inverted back through a multiplexed EX-OR gate at the D flip flop input. This permits efficient counters to be built using but a few gates to drive the state transitions.

The XC7318 macrocell is identical to the XC7336, but the XC7318 provides half of the number in the same package. Figure 4 shows the XC7318 architecture and Table 1 summarizes key supported features.

**Table 1. Key XC7336/XC7318 Features**

Feature	XC7336	XC7318
Fast Function Blocks	4	2
Fast Inputs	12	12
Fast Clocks	2	2
Fast Output Enables	2	2
3.3 V/5.0 V I/O	Yes	Yes
T <sub>pd</sub> (Fast In to Fast Out)	5.0/7.5/10/12 ns	5.0/7.5 ns
T <sub>pd</sub> (I or I/O to Outputs)	8.5 ns	8.5 ns
F <sub>max</sub>	167 MHz	167 MHz
Number of I/Os	38	38
Current Drive	24 mA	24 mA

Designers needing input registers, additional function block product terms, more macrocells and built in arithmetic operations should try the XC7354. The XC7354 offers more functionality than the XC7336/XC7318 and comes in 44-pin and 68-pin packages.

Table 2 shows the pin functionality of the XC7318, the XC7336 and the XC7354. The righthand column summarizes the common functions of all three chips. By using the recommended pin assignment on the righthand side, for 44-pin designs, an important capability can be had. Specifically, designs can be simply migrated among the 3 parts. For instance, designs wanting minimum density can use the XC7318. Greater density with equivalent speed can be gained by using the XC7336 and still greater density can be had by using the XC7354. Should a design initially target the XC7318, the exact same design can be moved into the other parts, if additional capability is required.

### Timing

Figure 5 shows the timing model for the XC7336/XC7318. Multiple paths are represented from input pin to output pin, depending upon whether signals are assigned to pass through the UIM, bypass the D flip flop, incorporate product term exporting and so forth. There are relatively few combinations, but each signal should be tallied separately for its pin to pin time delay, as needed. As an example, tally the time delay of three different signals:

1. Pin-to-pin, fastest path:

$$T_{delay} = t_{in} + t_{flogi} + t_{fpdi} + t_{fout}$$

2. Pin-to-Pin delay through the UIM:

$$T_{delay} = t_{in} + t_{uim} + t_{flogi} + t_{fpdi} + t_{fout}$$

3. Pin-to-Pin with two levels of p-term exporting:

$$T_{delay} = t_{in} + t_{flogi} + 2 t_{ptxi} + t_{fpdi} + t_{fout}$$

### Design Techniques

#### Automatic Software

The design examples presented are shown using PLUS-ASM which is simple and easy to understand. Typically, designers don't need to designate specific function mapping into the XC7336/XC7318, but occasionally, designers like to control how a solution is implemented, so these methods may be of interest.

Boolean operators used by PLUSASM are simply /, + and \* for INVERT, OR and AND, respectively. Combinational logic expressions are formed with a simple equal (=) sign, with operands and operators located on the right hand side of the expression. It is possible to use the / operator on the left hand side of an equation, for brevity, when needed.

Flip flop expressions are formed by writing expressions for the specific control pins of the flip flop. The D-input is a special case, where the equal sign is replaced by the compound symbol :=. Clock inputs are determined by the syntax FFname.clkf, and reset inputs are designated by FFname.rstf.

Table 2. Recommended 44-Pin Interchange

Pin Number	XC7336	XC7318	XC7354	Recommended
1	I/FO/FI/(MR)	I/FI/(MR)	I/FI/(MR)	I/(MR)
2	I/FO/FI	I/FI	I/FI	I/FI
3	I/FO/FI	I/FI	I/FI	I/FI
4	I/FO/FI	I/FI	I/FI	I/FI
5	FO/FCLK0	FCLK0	O/FCLK0	FCLK
6	FO/FCLK1	FCLK1	O/FCLK1	FCLK
7	I/FO/FI	I/FO/FI	I/FI	I/FI
8	I/FO	I/FO	I/FO	I/FO
9	I/FO	I/FO	I/FO	I/FO
10	GND	GND	GND	GND
11	I/FO	I/FO	I/FO	I/FO
12	I/FO	I/FO	I/FO	I/FO
13	I/FO	I/FO	I/FO	I/FO
14	I/FO	I/FO	I/FO	I/FO
15	I/FO	I/FO	I/FO	I/FO
16	I/FO	I/FO	I/FO	I/FO
17	I/FO	I	I/O/FI	I
18	I/FO/FI	I/FI	I/O/FI	I
19	I/FO/FI	I/FI	I/O/FI	I
20	I/FO/FI	I/FI	I/O/FI	I/O
21	VCCINT	VCCINT	VCCINT	VCCINT
22	I/FO	I	I/FI	I/O
23	GND	GND	GND	GND
24	I/FO	I	I/FO	I
25	I/FO	I	I/FO	I
26	I/FO	I	I/FO	I
27	I/FO	I	I/FO	I
28	I/FI	I/FI	I/FI	I/FI
29	I/FO	I/O	I/FO	I/O
30	I/FO	I/O	I/FO	I/O
31	GND	GND	GND	GND
32	VCCIO	VCCIO	VCCIO	VCCIO
33	I/FO	I/O	I/FO	I/O
34	I/FO	I/O	I/FO	I/O
35	I/FO	I/O	I/FO	I/O
36	I/FO	I/O	I/FO	I/O
37	I/FO	I/O	I/FO	I/O
38	I/FO	I/O	I/FO	I/O
39	FO/FOE1	O/FOE1	O/(CKEN0)	O
40	FO/FOE0	FOE0	O/FOE0	FOE
41	VCCINT	VCCINT	VCCINT/VPP	VCCINT
42	I/FI	I/FI	I/FI	I/FI
43	I/FO/FI	I/FO/FI	I/FI	I/FI
44	I/FO/FI	I/FO/FI	I/FI	I/FI

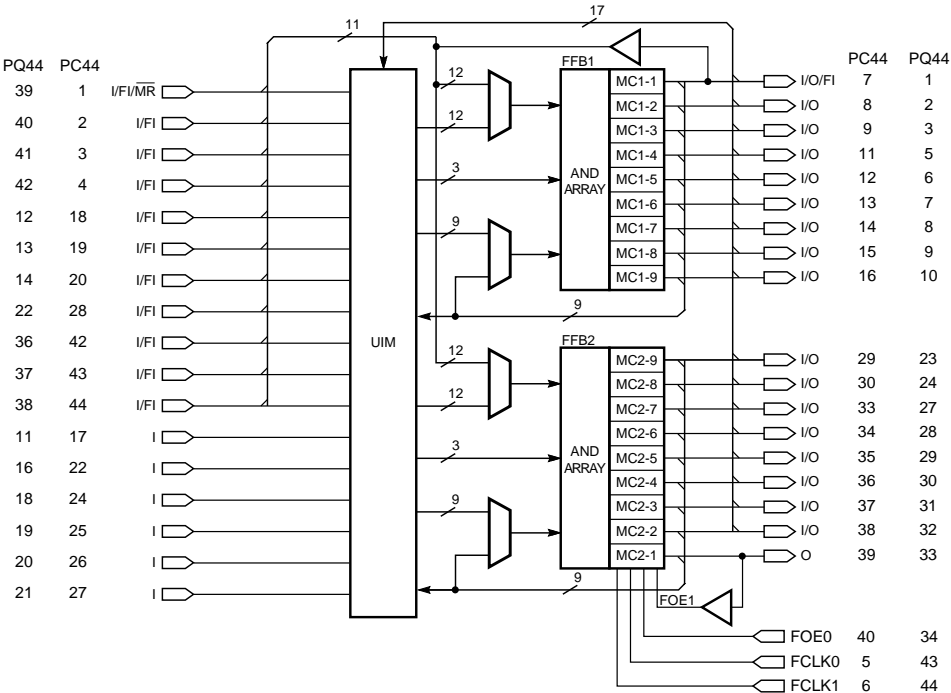
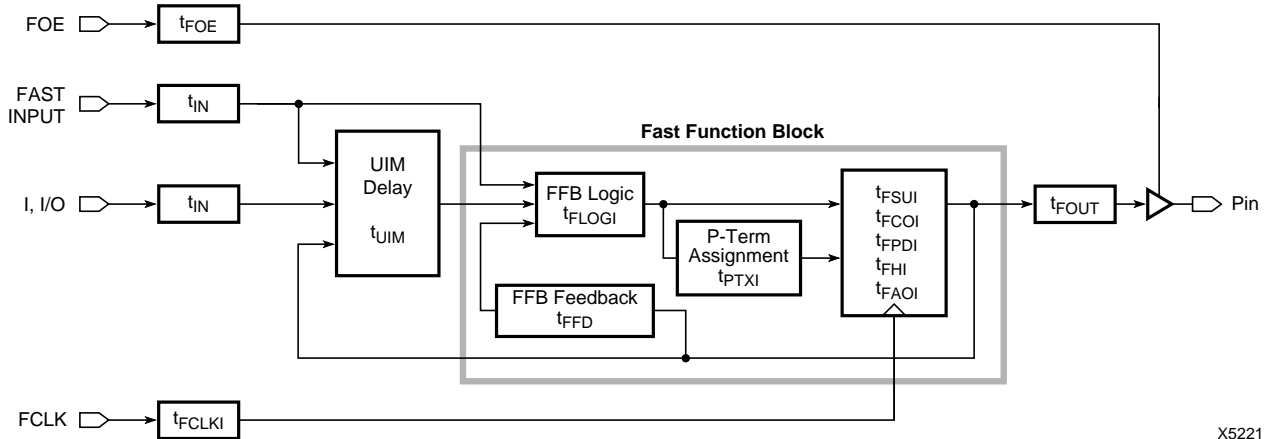


Figure 4. XC7318 Block Diagram

X5455



X5221

**Figure 5. XC7336/XC7318 Timing Model**

A design file contains a header section including optional documentation sections and mandatory declaration of inputs, outputs, global signals and any user preferred arrangement of functions. The interested reader is referred to the 1994 XEPLD Design Guide for further details.

### SMARTswitch™

SMARTswitch is the automatic XEPLD software that places logic into the UIM. As single lines enter the UIM, they are assigned to function block inputs, making connections. However, multiple lines driving a UIM connection form a wired AND function, that is useful for making additional logic. Forming UIM logic will be illustrated in the Counter and Comparator examples shown later.

### Gates

The following expressions show the basic logic operations.

$$\begin{aligned} \text{ABAR} &= /A; \\ \text{AORB} &= A+B; \\ \text{AANDB} &= A*B; \\ / \text{ANORB} &= A+B; \\ / \text{ANANDB} &= A*B; \\ \text{AEXORB} &= /A*B + A*/B; \\ \text{AEXNORB} &= A*B + /A*/B; \end{aligned}$$

### Muxes and Decoders

Using the above methods, compound expressions are formed to build up logic functions. Using A0 to A3, B0 to B3 and SEL (select) as inputs, a multiplexer follows:

$$\begin{aligned} \text{DAT0} &= \text{SEL} * \text{A0} + / \text{SEL} * \text{B0}; \\ \text{DAT1} &= \text{SEL} * \text{A1} + / \text{SEL} * \text{B1}; \\ \text{DAT2} &= \text{SEL} * \text{A2} + / \text{SEL} * \text{B2}; \\ \text{DAT3} &= \text{SEL} * \text{A3} + / \text{SEL} * \text{B3}; \end{aligned}$$

The approach extends to larger multiplexers. The above uses one macrocell per data bit, and leaves behind two unused product terms in each macrocell. To take advantage of four product terms per macrocell, the idea expands as follows:

$$\begin{aligned} \text{DAT0} &= \text{S1} * \text{S0} * \text{D0} + \text{S1} * / \text{S0} * \text{C0} + / \text{S1} * \text{S0} * \text{B0} + / \text{S1} * / \text{S0} * \text{A0} \\ \text{DAT1} &= \text{S1} * \text{S0} * \text{D1} + \text{S1} * / \text{S0} * \text{C1} + / \text{S1} * \text{S0} * \text{B1} + / \text{S1} * / \text{S0} * \text{A1} \\ \text{DAT2} &= \text{S1} * \text{S0} * \text{D2} + \text{S1} * / \text{S0} * \text{C2} + / \text{S1} * \text{S0} * \text{B2} + / \text{S1} * / \text{S0} * \text{A2} \\ \text{DAT3} &= \text{S1} * \text{S0} * \text{D3} + \text{S1} * / \text{S0} * \text{C3} + / \text{S1} * \text{S0} * \text{B3} + / \text{S1} * / \text{S0} * \text{A3} \end{aligned}$$

Very high speed decoders can be built in the macrocells, to form SRAM select signals, but do not use all of the macrocell product terms or the flip flop in most cases. Decoders are formed as follow:

$$\begin{aligned} \text{DEC0} &= / \text{A3} * / \text{A2} * / \text{A1} * / \text{A0} \\ \text{DEC1} &= / \text{A3} * / \text{A2} * / \text{A1} * \text{A0}; \\ \text{DEC2} &= / \text{A3} * / \text{A2} * \text{A1} * / \text{A0}; \end{aligned}$$

Using Equation 1 from the Timing section, we can determine the time delay required to generate DEC0,1 or 2 as follows for the XC7318-5:

$$\begin{aligned} t_{in} &= 1.5 \text{ ns} \\ t_{flogi} &= 1 \text{ ns} \\ t_{fpdi} &= 0.5 \text{ ns} \\ t_{fout} &= 2.0 \text{ ns} \\ T_{delay} &= 1.5 + 1 + 0.5 + 2 = 5 \text{ ns} \end{aligned}$$

### Registers=

Simple registers are formed as follows:

$$\begin{aligned} \text{A} &:= \text{DATAINPUT}; \\ \text{A.CLKF} &= \text{CLOCK}; \\ \text{A.RSTF} &= \text{RESET}; \end{aligned}$$

This describes a D flip flop with its input tied to something named DATAINPUT, its clock tied to a signal called CLOCK and its reset input tied to a signal called RESET.

### Shift Registers

Cascading registers results in a shift register, as follows:

```
A:=DATAINPUT;
B:=A;
C:=B;
D:=C;
A.CLKF = CLOCK;
B.CLKF = CLOCK;
C.CLKF = CLOCK;
D.CLKF = CLOCK;
A.RSTF = RESET;
B.RSTF = RESET;
C.RSTF = RESET;
D.RSTF = RESET;
```

This shifter uses four macrocells, and if the signals designated A,B,C,D are declared as outputs, they will appear somewhere at the pins of an XC7336/XC7318. If A,B,C and D are declared as nodes (internal points), the software buries them.

### Counters

Counters can be built in a number of different ways. The most efficient is to have the macrocell flip flops configured as T flip flops. The following equations form T flip flops, add logic to load the flip flops, and include a term permitting the counter to be paused (CE = Count Enable):

```
Q0.T = LOAD */CE*D0*/Q0 +
LOAD*/CE*/D0*Q0 + /LOAD*CE
Q0.CLKF = CLOCK
Q0.RSTF = RESET

Q1.T = LOAD */CE*D1*/Q1 +
LOAD*/CE*/D1*Q1 + /LOAD*CE *Q0
Q1.CLKF = CLOCK
Q1.RSTF = RESET

Q2.T = LOAD */CE*D2*/Q2 +
LOAD*/CE*/D2*Q2 + /LOAD*CE*Q0*Q1
Q2.CLKF = CLOCK
Q2.RSTF = RESET

Q3.T = LOAD */CE*D3*/Q3 +
LOAD*/CE*/D3*Q3 + /LOAD*CE*Q0*Q1*Q2
Q3.CLKF = CLOCK
Q3.RSTF = RESET
```

```
Q4.T = LOAD */CE*D4*/Q4 +
LOAD*/CE*/D4*Q4 + /LOAD*CE*Q0*Q1*Q2*Q3
Q4.CLKF = CLOCK
Q4.RSTF = RESET
```

```
Q5.T = LOAD */CE*D5*/Q5 +
LOAD*/CE*/D5*Q5 + /LOAD*CE*Q0*Q1*Q2*Q*Q4
Q5.CLKF = CLOCK
Q5.RSTF = RESET
```

```
Q6.T = LOAD */CE*D6*/Q6 +
LOAD*/CE*/D6*Q6 + /LOAD*CE*Q0*Q1*Q2*Q3*Q4*Q5
Q6.CLKF = CLOCK
Q6.RSTF = RESET
```

```
Q7.T = LOAD */CE*D7*/Q7 +
LOAD*/CE*/D7*Q7 + /
LOAD*CE*Q0*Q1*Q2*Q3*Q4*Q5*Q6
Q7.CLKF = CLOCK
Q7.RSTF = RESET
```

```
Q8.T = LOAD */CE*D8*/Q8 +
LOAD*/CE*/D8*Q8 + /
LOAD*CE*Q0*Q1*Q2*Q3*Q4*Q5*Q6*Q7
Q8.CLKF = CLOCK
Q8.RSTF = RESET
```

Count bits Q0 through Q8 fit nicely into a function block. To cascade into the next function block, an additional term is needed, that passes the all one condition of the low order bits. The term is called CO.

Before adding in the CO term, the fastest frequency that a 9 bit counter can achieve will be calculated. This is derived from Figure 5 by tallying the logic time delay ( $t_{flogi}$ ), with the flip flop required times for setup ( $t_{su}$ ), clock to output ( $t_{coi}$ ) and the feedback time delay ( $t_{ffd}$ ). This results in a timing requirement as follows:

$$T_{\text{delay}} = t_{flogi} + t_{su} + t_{coi} + t_{ffd}$$

Using timing values for the 7.5 ns version of the XC7318 gives a calculation of the loop time delay as follows:

$$\begin{aligned} t_{flogi} &= 1.5 \text{ ns} \\ t_{su} &= 1.5 \text{ ns} \\ t_{coi} &= 1 \text{ ns} \\ t_{ffd} &= 4 \text{ ns} \\ T_{\text{delay}} &= 1.5 + 1.5 + 1 + 4 = 8 \text{ ns} \end{aligned}$$

The reciprocal of this is the fastest operating frequency the nine bit version of the counter can achieve:

$$F_{\text{max}} = 1/(8 \text{ ns}) = 125 \text{ MHz}$$

Adding in the CO term is done by logically ANDing through the UIM.

$$CO = Q0 * Q1 * Q2 * Q3 * Q4 * Q5 * Q6 * Q7 * Q8$$

$$Q9.T = LOAD * /CE * D9 / Q9 + LOAD * /CE * D9 * Q9 + /LOAD * CE * CO$$

$$Q9.CLKF = CLOCK$$

$$Q9.RSTF = RESET$$

$$Q10.T = LOAD * /CE * D10 / Q10 + LOAD * /CE * D10 * Q10 + /LOAD * CE * CO * Q9$$

$$Q10.CLKF = CLOCK$$

$$Q10.RSTF = RESET$$

In this design, the first 9 bits (Q0 - Q8) reside within an XC7336/XC7318 function block. Each flip flop toggles on the clock when its CE signal is a logic one and all flip flops of lower position are logically one. The condition of all lower bits being logical one is decoded by the AND of the particular bits, that are broadcast among the macrocells within the function block. When nine bits are reached in the counter design, a tenth bit must be conditioned for toggle by sending the term CO from one function block to another. The software automatically forms the AND gate logic for CO in the UIM and passes it as a single signal into the next function block. Figure 6 shows the UIM cascade with CO passed from one function block to another.

Timing for the expanded counter is altered by replacing the local feedback expression  $t_{FFD}$  with the time delay of a signal passing through the UIM,  $t_{UIM}$ . This value is 4.5 ns, taking the total loop delay to 8.5 ns resulting in a Fmax of 117 MHz.

The above counter design shows the regular pattern needed by an 11 bit up counter. The pattern is easily expanded for up to 36 bits in an XC7336 or 18 bits in an XC7318, and may be converted to a down counter by taking the /Q outputs outside and complementing the data being loaded. Note that two product terms are needed to load the flip flops, because T flip flops require both input data senses to load.

### Comparators

Comparators are easily handled by the XC7336/ XC7318 macrocell, but single bit comparators do not use all available macrocell product terms. A more efficient use is to handle four bits at a time to generate multiple compares per macrocell:

$$COMP = /B1 * /B0 * /A1 * /A0 + B1 * /B0 * A1 * /A0 + /B1 * B0 * /A1 * A0 + B1 * B0 * A1 * A0$$

Next, several COMP signals can be gated together to detect equality across larger groups of bits. Each group of four bits uses 4 function block inputs, meaning six four bit compares can occur per function block. Another macrocell or the UIM then forms the composite function of all the bit compares, as needed.

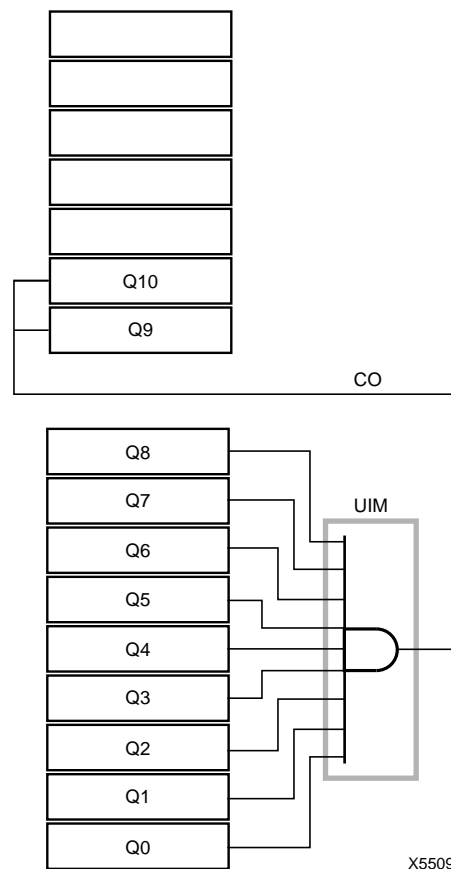


Figure 6. UIM Counter Cascade

An efficient comparator can be built using a slightly different strategy, appropriate for microprocessor address decoding. Assume a 12 bit operand (A0 - A11) resides in other function blocks than the compare function block. This value is registered, and may be compared very quickly to another 12 bit operand (B0 - B11) applied to the fast inputs, forming a low asserted compare signal at the function block output pin. The scheme works using the logical OR of multiple Exclusive OR functions. Each macrocell uses all four product terms, with automatic product term cascading. This approach takes 11 ns on the XC7336-7 part, and the output /COMP signal can be tied directly to chip enables on external memory chips. See Figure 7.

$$\begin{aligned} /COMP = & A0 * /B0 + /A0 * B0 \\ & + A1 * /B1 + /A1 * B1 \\ & + A2 * /B2 + /A2 * B2 \\ & + A3 * /B3 + /A3 * B3 \\ & + A4 * /B4 + /A4 * B4 \\ & + A5 * /B5 + /A5 * B5 \\ & + A6 * /B6 + /A6 * B6 \\ & + A7 * /B7 + /A7 * B7 \\ & + A8 * /B8 + /A8 * B8 \\ & + A9 * /B9 + /A9 * B9 \\ & + A10 * /B10 + /A10 * B10 \\ & + A11 * /B11 + /A11 * B11 \end{aligned}$$

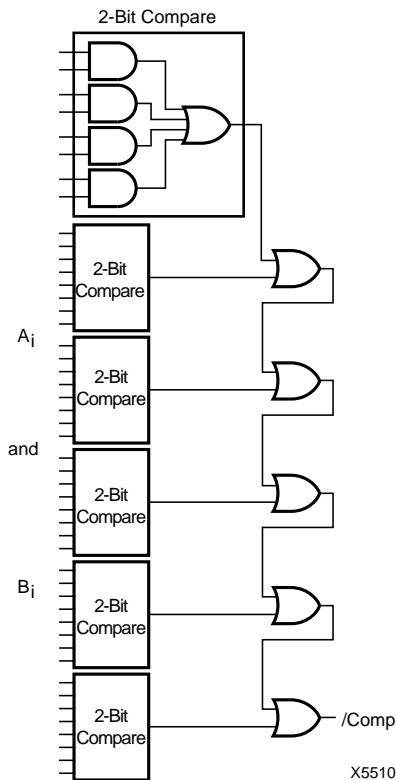


Figure 7. 12-Bit Comparator

### Latches

Occasionally, designers need a transparent latch within an XC7336 or XC7318. The latch is formed by feeding the macrocell combinatorial logic back upon itself per the following equation:

$$Q = \text{ENA} * \text{DATA} + \text{/ENA} * Q + Q * \text{DATA}$$

Note that the signal  $Q * \text{DATA}$  is included to eliminate a hazard, making  $Q$  glitch free.

### Merged Mux/Latches

Latches often occur when driven from multiplexers. When this happens, the MUX can be embedded right into the latch making a combined function that is faster than the stacked function and more efficient. The following equation merges a 2 to 1 multiplexer into the latch operation.

$$Q = \text{DAT1} * \text{SEL} * \text{ENA} + \text{DAT0} * \text{/SEL} * \text{ENA} + Q * \text{/ENA}$$

### Clock Tricks

Building latches is one way of introducing an additional clock signal, but designers often need to introduce product term clocks. P-term clocks can be formed by simply building a logic expression, which will be formed at a macrocell, and designating that logic expression as another flip flop's clock input. The software will automatically assign the clocking logic to the FCLK0 or FCLK1 pin, where the macrocell output is redirected back into the EPLD and assigned as designated.

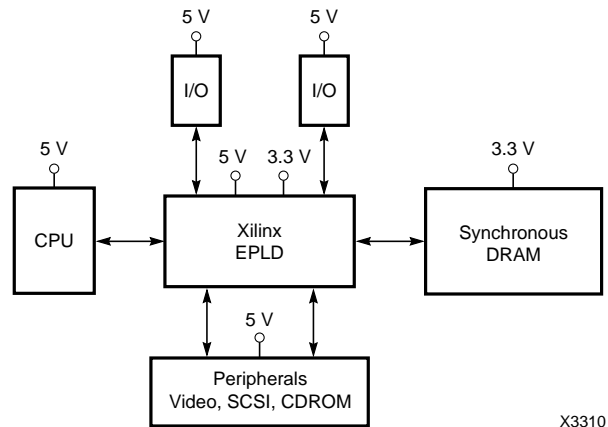


Figure 8. Typical Mixed Voltage System

## Practical Considerations for XC7336/XC7318 Designs

The XC7336/XC7318 offer additional capability not highlighted by the previous discussions. Systems using 3.3 volt and 5 volt devices are easily interfaced with the XC7336 / XC7318, by following a few simple rules. Also, very high speed EPLDs behave much better if standard high performance printed circuit board techniques are adhered to, so a small checklist is appropriate for those rules. And finally, best EPLD behavior is obtained by following a few guidelines with respect to the Master Reset (/MR) and power on Master Reset capabilities of the XC7336 and XC7318.

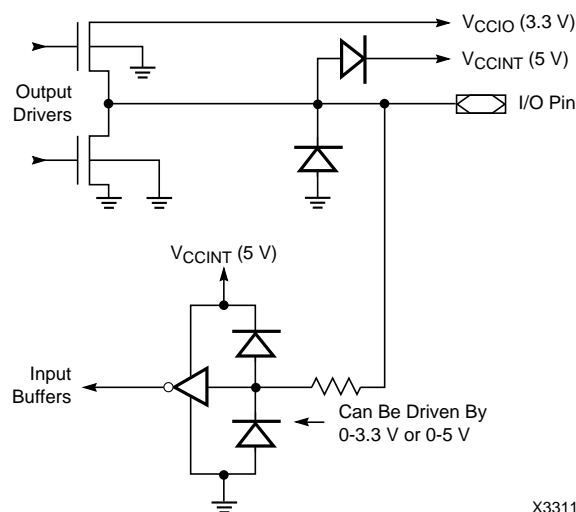
### Mixed Voltage Operation

Xilinx EPLDs support mixed voltage systems similar to that shown in Figure 8 combining both 3.3 and 5 Volt components. Xilinx EPLDs combine both logic and level shifting functions in a single programmable device, eliminating the need for discrete level translation buffers. These EPLDs feature split power supply rails. The internal core logic always runs at 5 volts for the fastest possible performance. The output buffers can be powered by either 5 volts or 3.3 volts by connecting the I/O  $V_{CC}$  to a 3.3 volt or 5 volt supply. True TTL compatibility allows the EPLDs to drive and be driven by any combination of 3.3 and 5 volt logic without any performance penalty, even when the I/O  $V_{CC}$  pins are powered by 3.3 volts.

The Xilinx EPLD I/O structure is shown in Figure 9. Input protection diodes connect to the internal 5 volt power supply rail, not the output buffer supply rail. This allows the input to withstand a maximum voltage of 7 volts, even when the I/O power pins connect to 3.3 volts. Since both output transistors are N-channel devices, there is no parasitic diode to be forward biased if the output is 3-stated and a 5 volt device is driving the EPLD I/O pin.

This enables the EPLD to operate on a bus that includes both 3.3 V and 5 V devices.





**Figure 9. Xilinx EPLD I/O Structure**

Mixed voltage power supplies may ramp-up simultaneously, or in either order. In all CMOS devices, current flows through the input pin protection circuitry if pins are driven before  $V_{CCINT}$  is applied. In the XC7336/XC7318, no damage will occur if the input current is less than 150 mA/pin. However, the user is advised to power-up all system power supplies simultaneously, thereby minimizing input transient currents.

Xilinx EPLDs are TTL-compatible with 3.3 and 5 volt logic as shown in Figure 10. The 5 volt TTL logic input thresholds are  $V_{IH} = 2.0$  V and  $V_{IL} = 0.8$  V. Xilinx EPLDs drive HIGH greater than 2.4 V and LOW below 0.4 V at rated output drive currents, with at least 400 mV noise margin.

### High Speed Design Considerations

The XC7336/XC7318 are offered in both 7.5 and 5 nanosecond versions. These very high speed parts have guaranteed maximum time delays that are 7.5 and 5 nanoseconds, and actual parts may in fact be somewhat faster. Because of this speed, additional care should be taken when using these parts, so that adjoining chips will operate properly.

Many high speed designs also require high current drive outputs for handling capacitive loads. The XC7336/XC7318 provide 24 mA drivers to eliminate the need for additional buffering that would decrease their speed. This results in a need to manage the total current being switched, so a strategy to do that is provided.

As with other high speed logic devices, the XC7336/XC7318 should use low inductance capacitors located as close as possible to the XC7336/XC7318  $V_{CC}$  and GND pins when mounted on a PC board. Care should be taken to mount the devices so that the PC interconnect traces are as close as possible to the target signal destinations.

### Layout Checklist:

Complying with the following checklist should assure a successful design with an XC7336 / XC7318:

1. Tie unused inputs (except master reset) to ground. An unused master reset should be tied high. Do not use the -u option in XEPLD.
2. Locate XC7336 / XC7318 parts near chips they drive or are driven from to minimize transmission line effects.
3. Use wide spacing between fast signal lines (particularly clocks) to minimize crosstalk.
4. Power pins ( $V_{CC}$  and GND) are recommended to be placed on separate printed circuit board planes. Fast signals should reside on another plane, as well.
5. Decouple the chip  $V_{CC}$  with a 0.1 microfarad capacitor directly connecting each physical chip  $V_{CC}$  to the nearest ground plane. Low inductance, surface mounted capacitors are recommended.
6. Decouple the printed circuit board power inputs with 0.1 uF ceramic (high frequency) and 100 uF electrolytic (low frequency) filter capacitors.
7. All device ground pins must be connected together.
8. Avoid using sockets to attach XC7318 and XC7336 parts to the PCB. Direct soldered connection minimizes inductance and reduces ground rise.

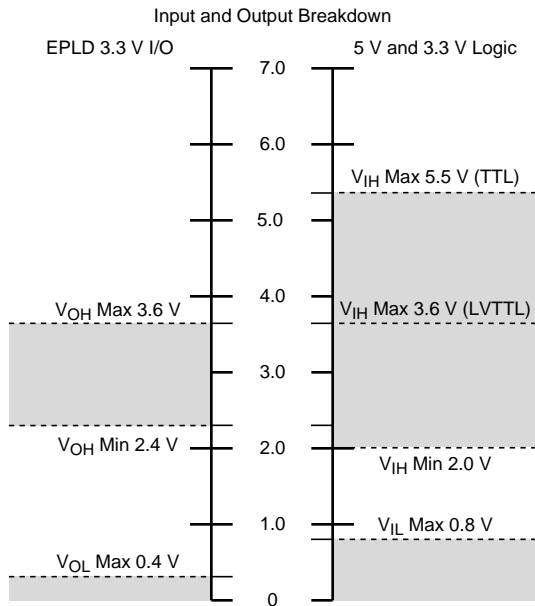
### Managing Ground Rise

Today's high performance designers must also be aware of additional factors that can affect the performance of fast, high current drive systems. As mentioned earlier, possible voltage rise on the ground pins of a device can affect the output levels being driven as well as sensed by the switching EPLD.

Figure 11 shows how ground rise is typically observed with today's high performance EPLDs. In this setup, multiple outputs are switched with a control variable, while one output is constantly being driven low and observed. As the multiple outputs switch, their in rushing current converges at the ground pin(s) of the EPLD. Lead impedance causes the reference ground to develop a voltage higher than before the switching outputs occurred. The result is that the static output being observed also develops an observable voltage swing.

All digital ICs have this property, and it causes no harm to the system unless the voltage swing on the static output is capable of switching another circuit down the line. Problems can occur if the voltage swing is excessive and this effect is particularly significant if the static (quiet) signal is attached to another circuit's clock input.

There are at least two factors that contribute to this ground rise. First, the amount of capacitive load being



X3317

**Figure 10. Xilinx EPLD Driving 3.3 V and 5 V Components**

driven is important because charge on this capacitance is the source of the in rushing current. Second, the number of simultaneous switching outputs is a factor because each switching output adds to the total capacitance being discharged.

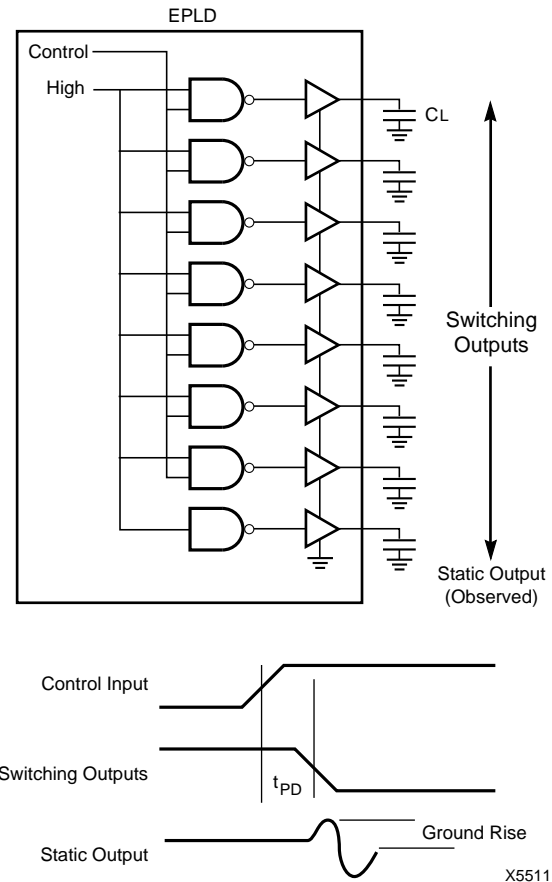
Unlike typical fast PLDs (shown switching in Figures 12 and 13), Xilinx EPLDs are not supplied in DIP packages with only a single ground. Xilinx EPLDs are supplied in symmetric packages that minimize lead inductance and supply multiple ground pins.

The XC7336 and XC7318 each have three grounds. Additional grounds and superior packages permit more outputs to switch simultaneously than other fast PLDs.

Figure 14 a and b show the XC7336-5 switching various numbers of outputs with a statically observed output responding similarly to those shown in Figures 12 and 13. Note that Figure 14 a and b have each output driving an 80 pF load, and that the small ground rise on the “static” output is less than that of Figures 11 and 12 where fewer outputs, driving less load capacitance result in greater ground rise on the static output. In summary, for AMD the ground rise is 1.66 V, for Lattice it is 2.0 V but the XC7336-5 has only 0.72 V.

The following checklist will reduce unnecessary ground rise:

1. Pinout only essential outputs. Intermediate shifter bits, and counter bits that need not drive outputs should remain buried.



X5511

**Figure 11. Ground Rise Test**

2. Minimize the number of outputs switching simultaneously. This can be accomplished by skewing the output enable signals FOE0 and FOE1. One way to do this is to simply route an enable to FOE0, and pass the same enable signal through a macrocell for delay, to drive FOE1. This is shown in Figure 15 and results in one TPD of time delay.

3. The two fast clock inputs can be managed similar to the two FOE signals, by delaying one of the clocks to gain signal skew.

4. Additional grounding can lower ground rise effects, and may be dealt with simply. Unused outputs can be tied directly to the PCB ground and driven low using internal logic. This splits the current driven into heavily loaded ground pins and lowers the voltage rise.

#### Master Reset and Power On Master Reset

The XC7336/XC7318 devices undergo a short internal initialization sequence upon device powerup. During this time ( $t_{RESET}$ ), the outputs remain 3-stated while the device is configured from its internal EPROM array and all registers are initialized. If the /MR pin is tied to  $V_{CCINT}$ ,

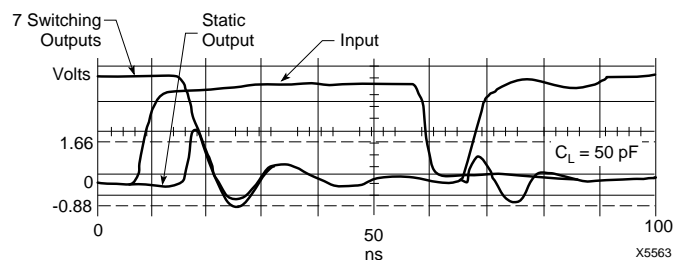


Figure 12. AMD PALCE 16V8H-7

the initialization sequence is completely transparent to the user and is completed in  $t_{\text{RESET}}$  after  $V_{\text{CCINT}}$  has reached 4.75 V. If  $\text{/MR}$  is held low while the device is powering up, the internal initialization sequence begins and the outputs will remain 3-stated until the sequence is complete and  $\text{/MR}$  is brought HIGH.  $V_{\text{CC}}$  rise must be monotonic to insure the initialization sequence is performed correctly.

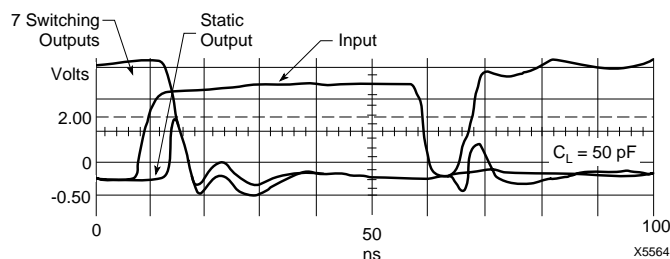


Figure 13. Lattice GAL 16V8-10

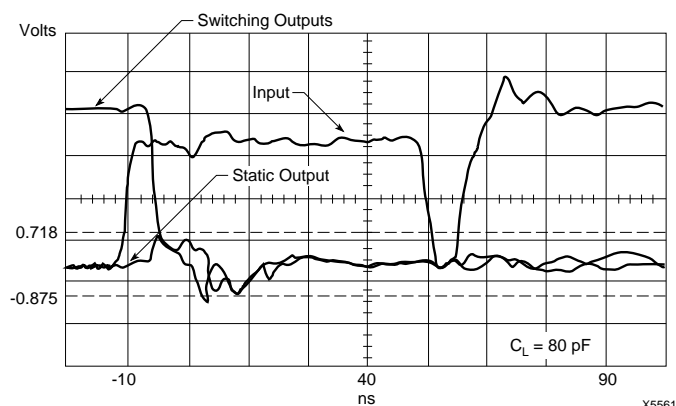


Figure 14a. XC7336-5 (Seven Switched Outputs)

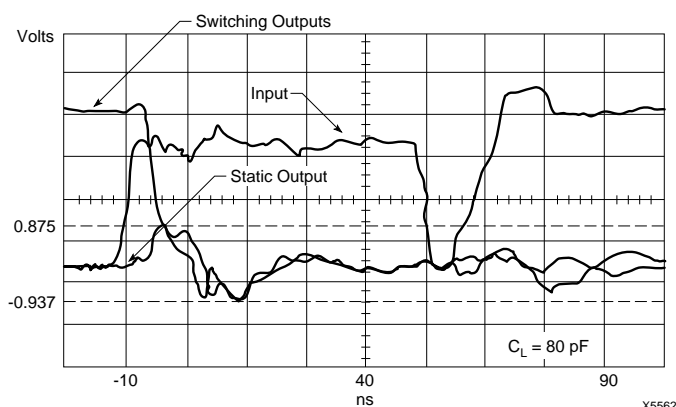


Figure 14b. XC7336-5 (12 Switched Outputs)

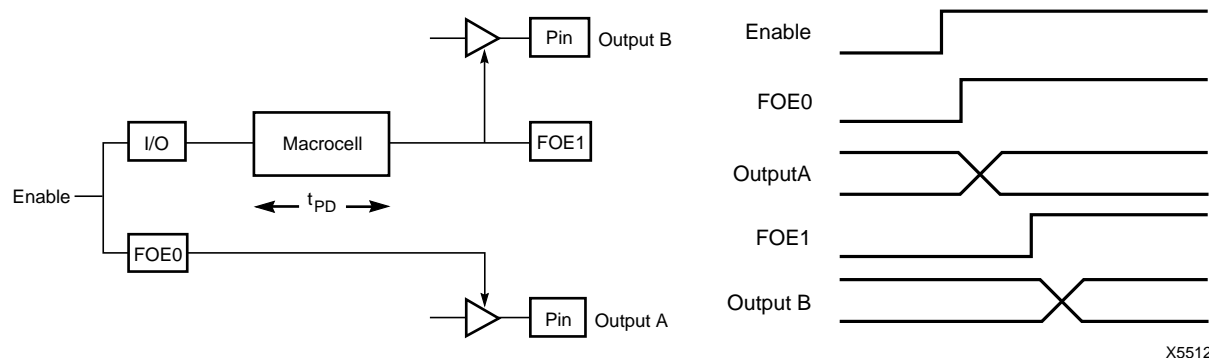
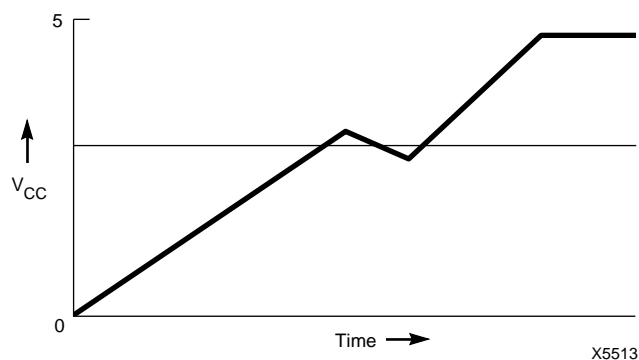


Figure 15. Output Enable Skewing Technique

Figure 16 shows a nonmonotonic rising supply voltage, that can adversely affect an XC7336/XC7318 during its power on initialization sequence.

Figure 16. Nonmonotonic  $V_{CC}$ 

For additional flexibility, the /MR pin is provided so the EPLD can be reinitialized after power is applied. On the falling edge of /MR, all outputs become 3-stated and the initialization sequence is started. The outputs will remain 3-stated until the internal initialization sequence is complete and /MR is brought HIGH. The minimum /MR pulse is  $t_{WMR}$ . If /MR is brought high after  $t_{WMR}$ , but before  $t_{RESET}$ , the outputs will become active after  $t_{RESET}$ . If /MR is used, it must be driven from a 5 V signal.



## Sales Offices

## Corporate Headquarters

Xilinx, Inc.  
2100 Logic Drive  
San Jose, CA 95124 U.S.A.  
Tel: 1 (408) 559-7778  
FAX: 1 (408) 559-7114

## Europe

Xilinx, Ltd.  
Suite 1B, Cobb House  
Oyster Lane  
Byfleet  
Surrey KT14 7DU  
United Kingdom  
Tel: (44) 932-349401  
FAX: (44) 932-349499

## Japan

Xilinx K. K.  
Daini-Nagaoka Bldg. 2F  
2-8-5, Hatchobori Chuo-ku.  
Tokyo 104, Japan  
Tel: (81) 3-3297-9191  
FAX: (81) 3-3297-9189