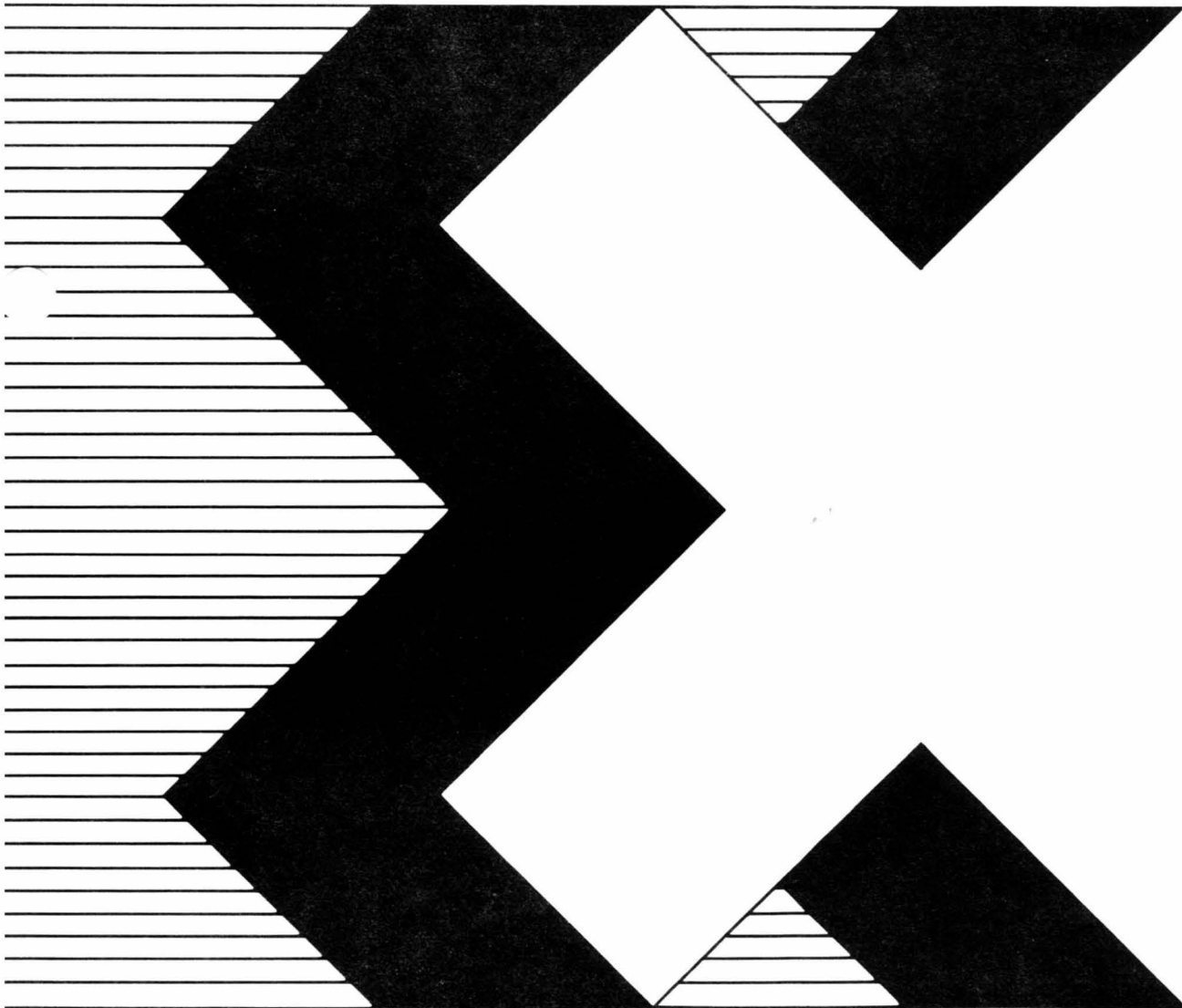


APPLICATION NOTE





Ins and Outs of Logic Cell™ Array I/O Blocks

Table of Contents

Introduction	2-21
I/O Structures	2-22
Schmitt-Triggers	2-27
Oscillators	2-29
Registers and Shift Counters	2-32
Output Current	2-38

INTRODUCTION

This Application Note describes various uses for Input/Output Blocks within the XC2064 and XC2018 Logic Cell Arrays.

The architecture of the XILINX Logic Cell Array (LCA) provides great design flexibility in using inputs and outputs. Since the Input/Output Blocks (IOBs) in an LCA are not dedicated to any fixed logic, they may also be used for many things beyond simple inputs or outputs. Many designs will not use all of the IOBs available within the LCA. In such cases, the design engineer can build logic structures such as shift registers or Johnson counters in unused IOBs.

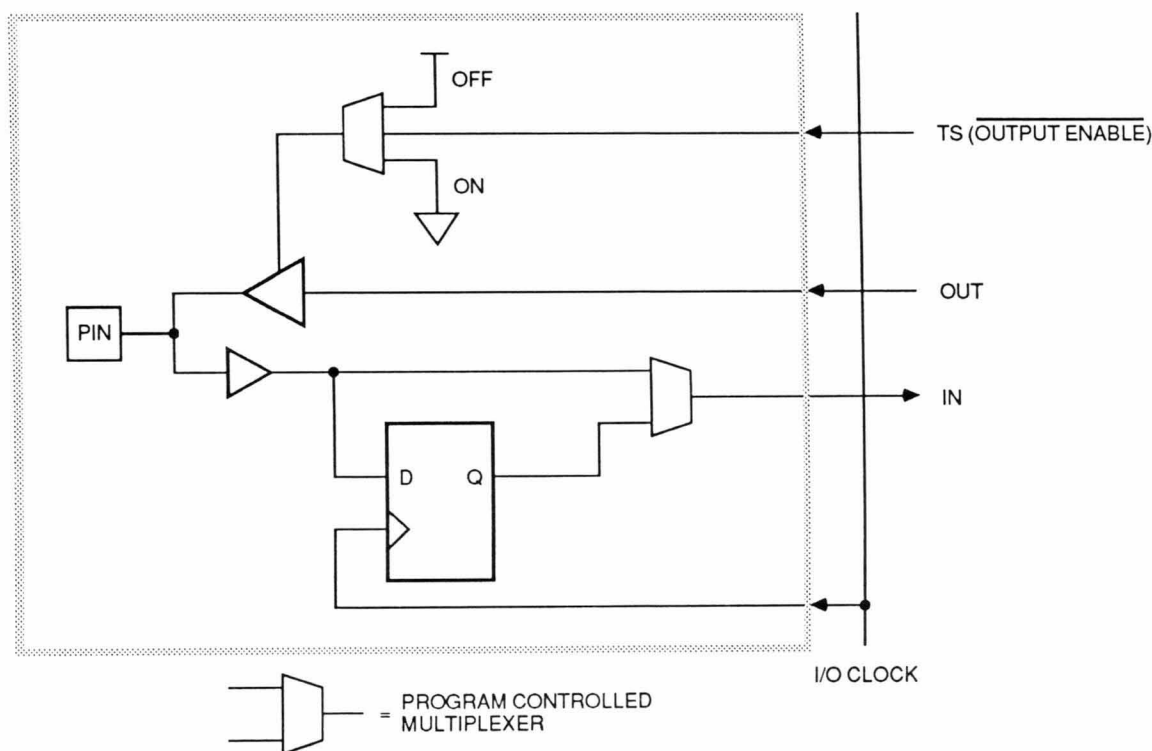
The Input/Output Block

All of the Input/Output Blocks (IOBs) of the Xilinx XC2064 and XC2018 Logic Cell Arrays (LCAs) are

identical. However, each IOB can be individually configured by the designer to perform a variety of logic functions. Each has the capability to drive an output, receive an input, clock the input into a flip-flop, or do both input and output under three-state control. A schematic of the IOB is shown in Figure 1. The trapezoidal structures are data path selectors or multiplexers. The programming of these data path selectors determines the function performed by the IOB.

Along each edge of the LCA die, the IOBs share a common I/O clock signal which drives the input register. All of the internal registers are reset to a "0" state after configuration or after the RESET pin is asserted low. Data is clocked into the input register on the positive edge of the I/O Clock signal.

Logic signals external to the LCA come in through an I/O pad and a non-inverting buffer. The signal is then either



0010022 1

Figure 1. Input/Output Block (IOB) . An Input/Output Block (IOB) can be configured as either a direct or registered input, a direct or three-state output, or as a bidirectional data line.

directly propagated or fed into the input register, depending on the configuration of the data path selector. Similarly, output data is driven by a non-inverting buffer. The output buffer is forced into a high-impedance state whenever the three-state control line is HIGH ($TS = 1$). Conversely, the output buffer propagates the output signal when the three-state control line is LOW ($TS = 0$). All outputs can source and sink 4 mA under specified worst-case conditions.

All IOBs can be globally configured to recognize either TTL-level ($V_{th} = 1.4$ Volts) or CMOS-level ($V_{th} = 2.2$ Volts) input thresholds. This option affects overall device power consumption. Power consumption is lower when CMOS input levels are selected.

Scope and Purpose

The purpose of this application note is to describe some functions available by configuring IOBs in various ways—some obvious, some not so obvious. These structures include:

- Standard Inputs and Outputs
- Open-Collector I/O
- Schmitt-Triggered I/O
- Oscillators
- IOB-based Registers (data, shift, etc.)
- Counters (Johnson, Linear Feedback)

I/O STRUCTURE DESCRIPTIONS

Conventions Used in the I/O Descriptions

For each I/O structure, the following conventions will be maintained. Each structure will be shown in schematic form and the IOB configuration will be described. Configurations available for the input path include:

- (I:PAD)—Direct input from the device pad.
- (I:Q)—Registered input.
- (I:)—No input.

The output buffer may be configured as follows:

- (BUF:ON)—Direct output.
- (BUF:TRI)—Three-state output.
- (BUF:)—No output.

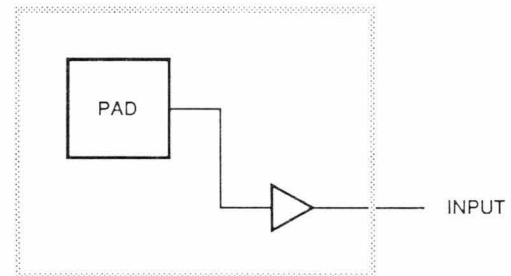
If a macro definition (MACRO) exists for the structure described, it will also be listed. Any special configuration information will be described under “COMMENTS.”

STANDARD I/O STRUCTURES

I/O TYPE: Pad Input

MACRO NAME: PIN

SCHEMATIC:



001022 2

CONFIGURATION:

I:PAD

BUF:

Figure 2. Pad Input

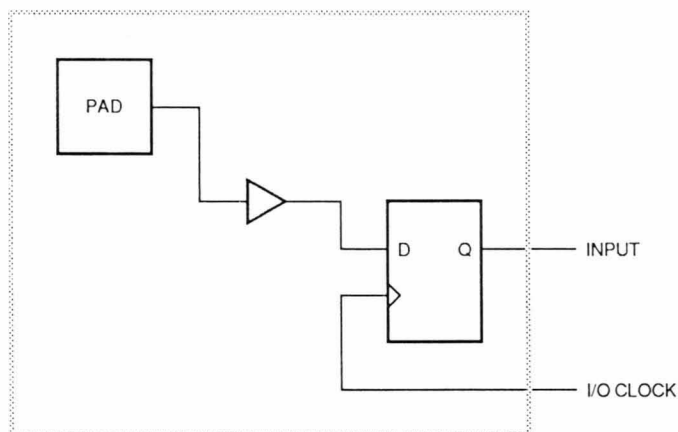
Registered Inputs and Metastability Characteristics

Figure 3 is a schematic of a registered input within the LCA. Logic Cell Arrays are manufactured with a high-speed CMOS process. This allows the I/O Block input registers to achieve flip-flop loop delays of three to five nanoseconds. Short loop delay provides very good performance under asynchronous clock and data transitions. Short loop delays minimize the probability of a metastable condition which can result when the input into the flip-flop is still in transition when the clock is asserted. The short loop delay characteristics of the I/O Blocks allow the device to be effective in synchronizing external signals. Once synchronized in the IOB, the signals can be used internally without further consideration of their relative timing, except as it applies to internal logic and routing path delays. Further information regarding the metastable behavior of registers within an LCA is contained in the XILINX application note, *Metastability Analysis of LCA Flip-Flops*.

I/O TYPE: Pad Input with Storage (registered input)

MACRO NAME: PINQ

SCHEMATIC:



0010028 3

CONFIGURATION:

I:Q

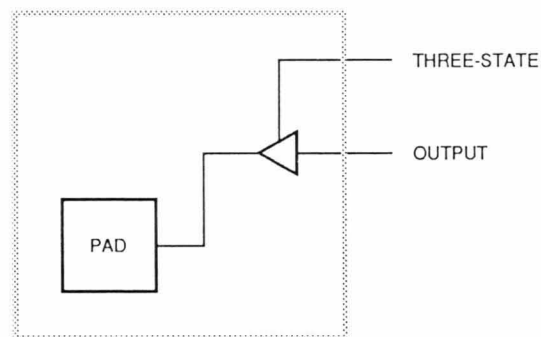
BUF:

Figure 3. Pad Input With Storage Register

I/O TYPE: Pad Output with Three-State Control

MACRO NAME: POUTZ

SCHEMATIC:



0010022 5

CONFIGURATION:

I:

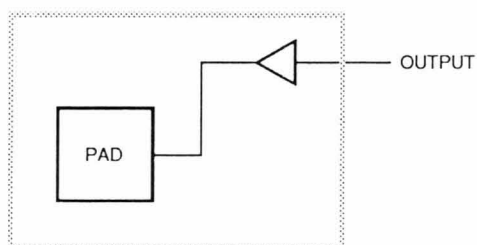
BUF:TRI

Figure 5. Pad Output With Three-State Control

I/O TYPE: Pad Output

MACRO NAME: POUT

SCHEMATIC:



0010022 4

CONFIGURATION:

I:

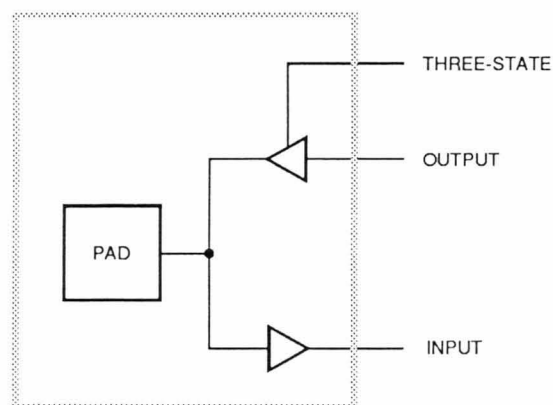
BUF:ON

Figure 4. Pad Output

I/O TYPE: Pad Input/Output (bidirectional data line)

MACRO NAME: PIO

SCHEMATIC:



0010022 6

CONFIGURATION:

I:PAD

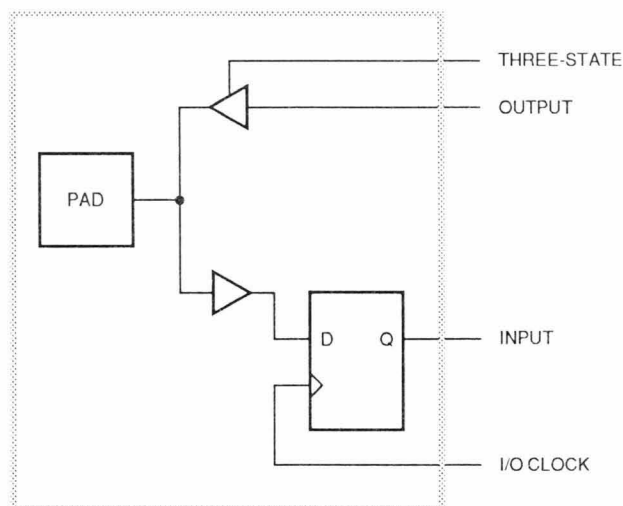
BUF:TRI

Figure 6. Pad Input/Output (Bidirectional)

I/O TYPE: Pad Input/Output with Input Storage

MACRO NAME: PIOQ

SCHEMATIC:



CONFIGURATION:

I:Q

BUF:TRI

0010022 7

Figure 7. Pad Input/Output With Input Storage

"OPEN-COLLECTOR" I/O STRUCTURES

Overview

"Open-collector" outputs can be used for a variety of functions including "wired" AND and OR structures. For MOS devices like the Logic Cell Array, a more accurate terminology is open-drain outputs, since an MOS transistor has no collector. To build an open-drain output structure in an LCA, both the output and the three-state control lines are tied together. For an active HIGH signal, the three-state control engages (high impedance) and the output signal is disabled through the output buffer. The signal at the output pad will also be high impedance, allowing that particular signal line to "float." Connecting this signal line to Vcc through a resistor will pull this line up for an active HIGH output. However, for active LOW signals, the three-state control line is driven LOW. This turns on the output buffer and allows the LOW signal to propagate directly to the I/O pad.

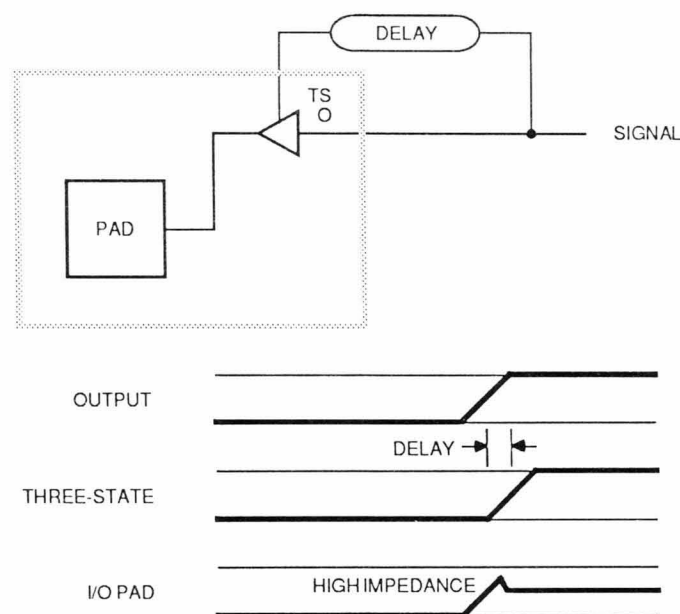
To build an open-drain output structure in an LCA, both the output and the three-state control lines are tied together. For an active HIGH signal, the three-state control engages (high impedance) and the output signal is disabled through the output buffer. The signal at the output pad will also be high impedance, allowing that particular signal line to "float." Connecting this signal line to Vcc through a resistor will pull this line up for an active HIGH output. However, for active LOW signals, the three-state control line is driven LOW. This turns on the output buffer and allows the LOW signal to propagate directly to the I/O pad.

Open-Drain Structures and Routing

When designing with open-drain structures, the designer should be aware of an LCA-specific phenomenon caused by the different routing delays between the signal source and the output and three-state control loads.

Since a routed signal may take longer to reach an IOB's three-state control line than its output line, the pad may be driven for a short period of time during a LOW to HIGH transition. This situation could occur if the output line (O) starts to go HIGH before the three-state control line (T). Depending on how much routing delay there is between the output (O) and three-state (T) lines, the PAD output could start to go HIGH and then be driven into high-impedance. Excessive routing delay differences between the output (O) and the three-state control line (TS) may cause a brief output glitch as shown in Figure 8. Careful design will prevent this.

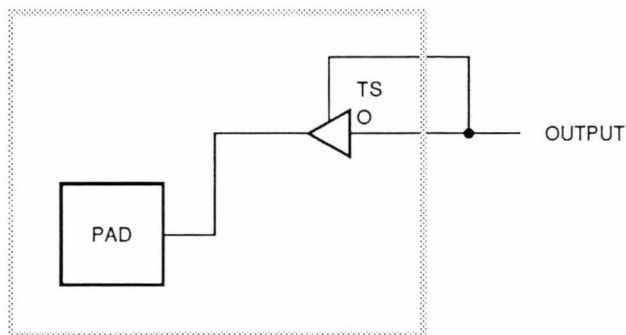
This situation is not a problem in most designs. The actual routing delay difference between the (T) and (O) terminals of an IOB can be checked using the timing calculator included in the XACT™ Development System.



0010022 8

Figure 8. Brief Output Glitch Caused by Three-State Routing Delay

I/O TYPE: Pad Output with “Open-Collector”
MACRO NAME: POUTC
SCHEMATIC:

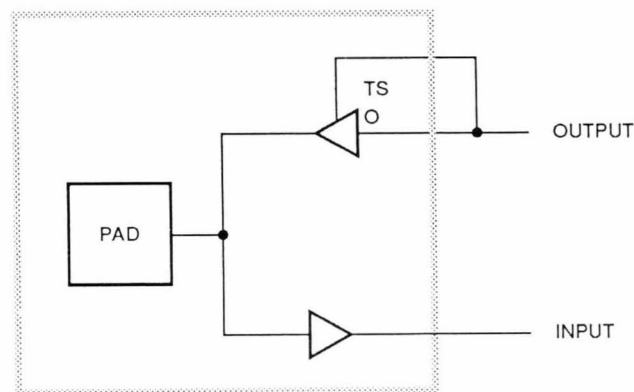


0010022 9

CONFIGURATION:
I:
BUF:TRI

Figure 9. Pad Output With “Open-Collector”

I/O TYPE: Pad Input/Output with “Open Collector”
MACRO NAME: PIOC
SCHEMATIC:

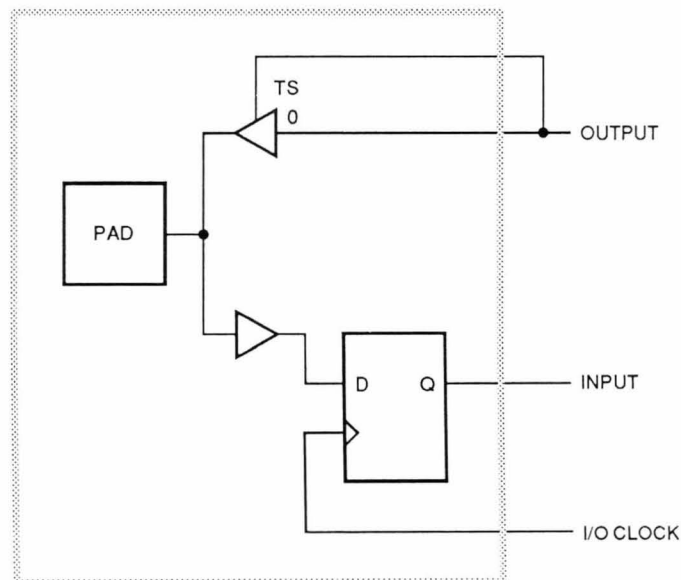


0010022 10

CONFIGURATION:
I:PAD
BUF:TRI

Figure 10. Pad Input/Output With “Open-Collector”

I/O TYPE: Pad Input/Output with Storage,
 “Open Collector”
MACRO NAME: PIOQC
SCHEMATIC:



0010022 11

CONFIGURATION:
I:Q
BUF:TRI

Figure 11. “Open-Collector” Pad Output With Storage

“Wired” AND and “Wired” OR Structures

The open-drain capability of Input/Output Blocks allows a designer to build “wired” AND and “wired” OR structures. The AND and OR implementations are essentially the same. The only difference between the forms is the type of logic used. “Wired” AND structures are used in positive-logic implementations, while “wired” OR structures are used in negative-logic implementations.

Figure 12 shows a typical “wired” AND or “wired” OR structure. All of the output PADs from the IOBs are externally wired together as a common signal. In a positive logic system, when all of the logic outputs to the

IOBs are true, the three-state control is enabled and the IOB output PADs are forced to high-impedance. However, since all of the IOBs are tied to Vcc through a pull-up resistor, the line is pulled-up to Vcc. If the logic signal to any of the IOBs is false, the corresponding output buffer would be turned on and that LOW signal would propagate to the common line. This, in turn, would pull the entire line LOW. The entire structure then acts as an AND function—when all outputs are high, the common line is high. If any output is low, then the common line is also low. The “wired” AND logic is

shown in Equation 1.

$$IOB1 \cdot IOB2 \cdot IOB3 \cdot \dots \cdot IOBn = \text{TRUE} \quad [1]$$

A “wired” OR structure is similar except that it is implemented in negative logic. It ORs together a number of active LOW signals to generate a logic function. The logic equation for a “wired” OR shown in Equation 2 structure is merely a “DeMorganized” inversion of Equation 1.

$$IOB1 + IOB2 + IOB3 + \dots + IOBn = \text{FALSE} \quad [2]$$

A typical application of a “wired” OR structure is an active low common interrupt line. If any peripheral requests an interrupt, the common interrupt line is pulled low, signalling the processor of the request. A “wired”-AND or “wired”-OR function can be built from any number of open-collector outputs.

Multiplexers from “Open Collector” outputs

Another structure which can be built using open drain IOBs is an n-bit multiplexer, as shown in Figure 13. All of the PAD outputs are tied together outside of the package on a common line which becomes the multiplexer output. Each IOB in the example is configured as an output with three-state control (Macro = POUTZ). The output line (O) of each IOB becomes an input for the multiplexer. A signal is selected by driving the corresponding three-state control line low (T = 0). The selected signal then propagates to the common output line. The three-state control lines can be driven with a Configurable Logic Block.

CAUTION: The designer must avoid output contentions on the common output line.

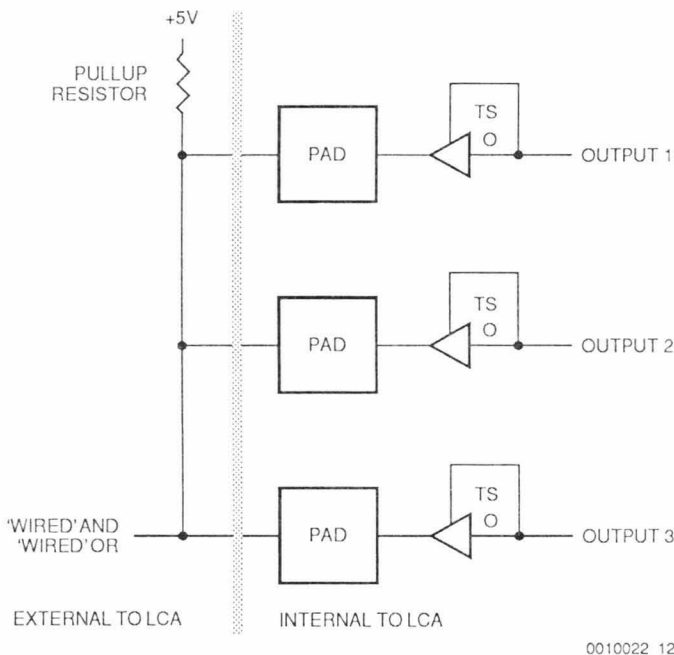


Figure 12. “Wired” AND or “Wired” OR Function

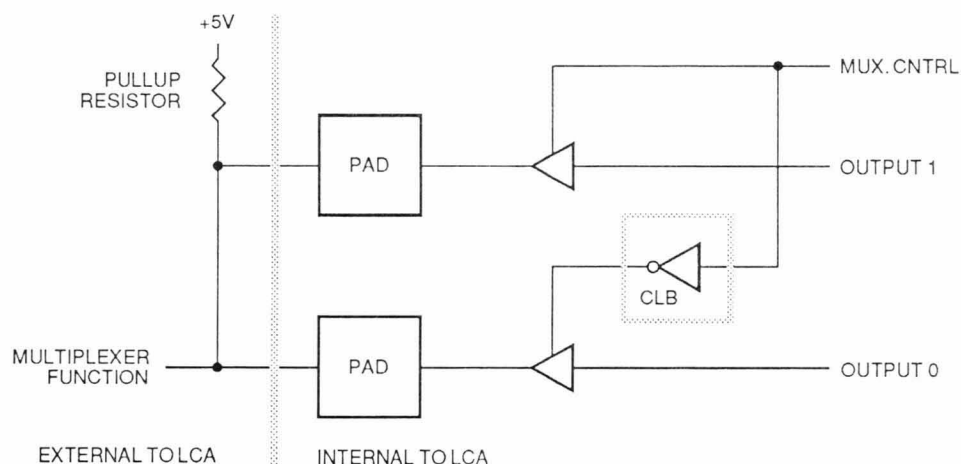


Figure 13. “Open Collector” Multiplexer Function. A multiplexer can be built from open-collector outputs. A specific output signal is selected by enabling the output buffer for that signal (TS = 0).

SCHMITT-TRIGGERED I/O STRUCTURES¹

Overview

The Schmitt trigger has numerous applications in digital designs. Two of the most common are shown in Figure 14. Schmitt-triggered inputs can filter signal noise due to hysteresis built into the switching characteristics. Schmitt triggers are also useful to generate fast transitions when a slowly changing input function reaches a predetermined level. Again, this effect is due to hysteresis.

Schmitt-triggered inputs can be built in a number of ways within an LCA. Using three IOBs, a CLB, and three resistors, a designer can build Schmitt triggers with selectable voltage hysteresis. If the amount of hysteresis is not critical, then the resource requirements are reduced to only two resistors and two IOBs.

The threshold voltage and the amount of hysteresis for a complete Schmitt trigger are selected using three resistors. The three resistors are separated into two resistor network pairs (R1:R2 and R1:R3) as shown in Figure 15. Each pair forms a voltage divider to set the input voltage level—one to set the HIGH going transition level (V_h) and one to set the LOW going transition level (V_l). The input value at the input to IOB1 is inverted through a CLB and then routed to the three-state control line IOB3. The CLB logic adds a small amount of time hysteresis to the signal since both the CLB logic

and the routing cause delay. The logic delay can be balanced by buffering the input before sending it to the three-state control of IOB2.

An inverting Schmitt trigger is similar except that the sense of the logic is inverted inside the LCA.

Theory of operation

Assume that the input voltage is near ground. The output voltage of IOB2 is at V_{ol} , which pulls resistor R2 toward ground. There is then no potential difference across R2. The output buffer of IOB3 is high-impedance, since its three-state control pin is HIGH. Resistor R3 is effectively removed from the circuit. The input voltage is divided by the resistor network formed by resistors R1 and R2. As the input voltage continues to increase, the IOB1 pad voltage will eventually reach its switching threshold.

Once the threshold is crossed, IOB1 goes HIGH, driving the output of IOB2 into high-impedance (IOB2 TS=1) and enabling the output buffer of IOB3 (IOB3 TS=0). Now at V_{oh} , IOB3 pulls the input of IOB1 high through resistor R3. In this state, resistor R2 is effectively removed from the circuit, since IOB2 is high-impedance.

This structure will remain in its present state even if the input voltage fluctuates. If the input voltage fluctuates to the opposite hysteresis limit, the Schmitt trigger will go to the opposite state. In other words, the Schmitt trigger will stay HIGH until the input to IOB1 drops below the LOW going hysteresis limit and *vice versa*.

If the values of the hysteresis are not critical, the Schmitt trigger can be reduced to only two IOBs and two resistors as shown in Figure 16. However, the range of V_h and V_l are very limited. The IOB configured as an output pulls the input HIGH or LOW, depending on the transition direction.

If a selectable Schmitt trigger is required only for a single transition direction (HIGH going LOW, or LOW going HIGH), then the circuit shown in Figure 15 can be further simplified to those shown in Figure 17a and 17b. Note, however, that a single CLB is required to invert the sense of input signal to enable or disable the output buffer for IOB2 (the IOB configured as a three-state output).

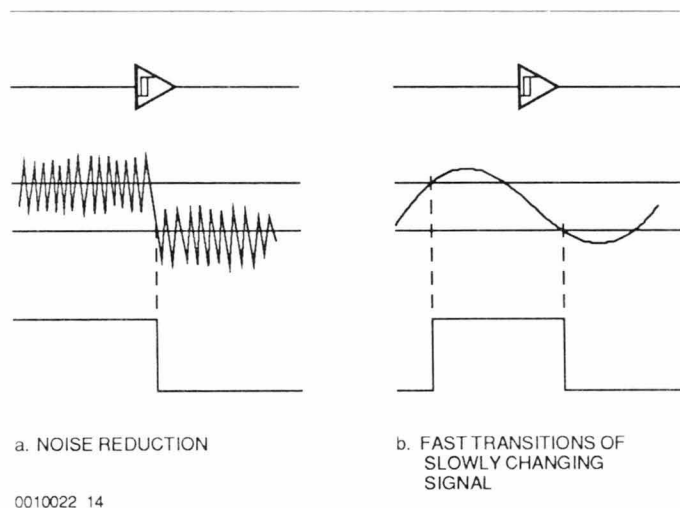
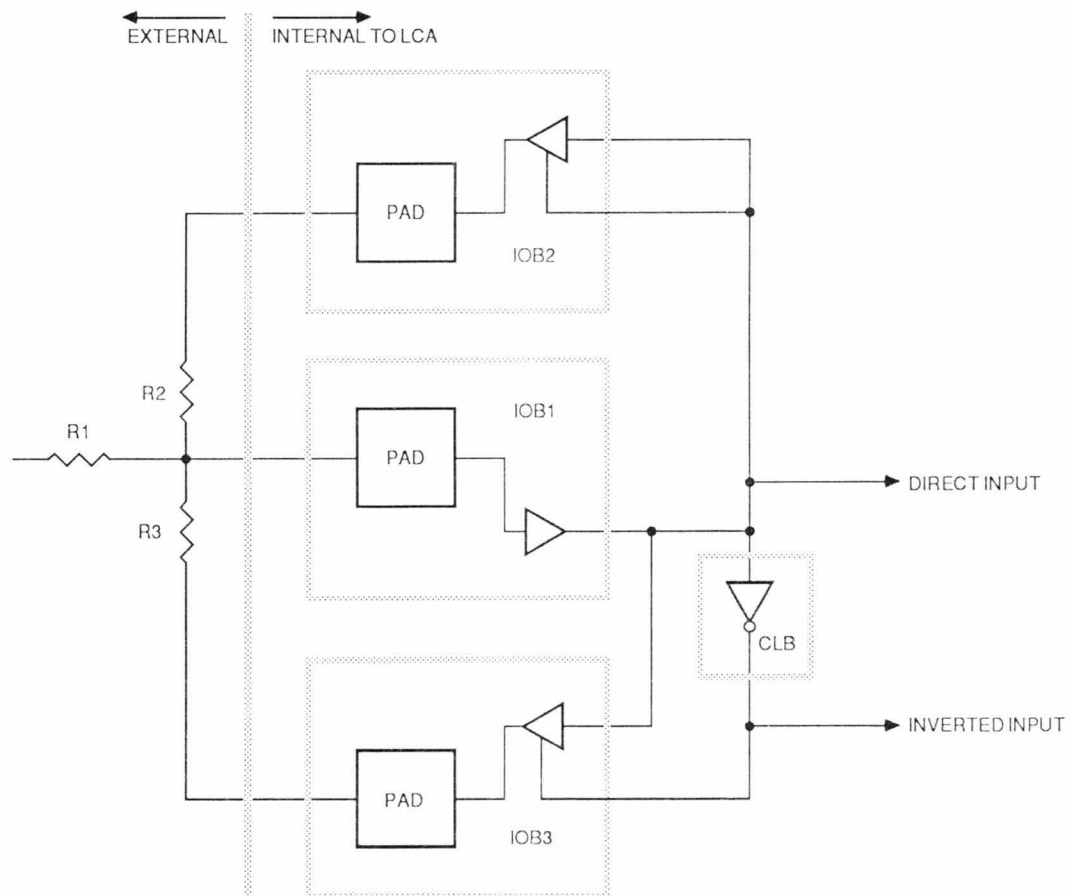


Figure 14. Application of a Schmitt trigger. Example applications of a Schmitt trigger include a.) A noisy input signal cleaned up through a Schmitt trigger. b.) Fast transitions generated from a slowly changing input signal through a Schmitt trigger input.

I/O TYPE: Schmitt-Triggered Input with Selectable Hysteresis
MACRO NAME: None
SCHEMATIC:



0010022 15

CONFIGURATION:

IOB 1 - Input

I:PAD

BUF:

IOB 2 - Output

I:

BUF:TRI

IOB 3 - Output (inverted through CLB)

I:

BUF:TRI

COMMENTS: Resistors pairs R1:R2 and R1:R3 form two voltage dividers which set the HIGH-going and LOW-

going input hysteresis. Resistors R1 and R2 set the HIGH-going hysteresis (V_h) according to Equation 3.

$$V_h = V_{th} \left[\frac{(R1+R2)}{R2} \right] - V_{ol} \left(\frac{R1}{R2} \right) \quad [3]$$

Resistors R1 and R3 set the LOW-going hysteresis (V_l) according to Equation 4.

$$V_l = V_{th} \left[\frac{(R1+R3)}{R3} \right] - V_{oh} \left(\frac{R1}{R3} \right) \quad [4]$$

Notes: V_{th} = input threshold voltage
 for CMOS inputs, $V_{th} = 2.2 \text{ V}$
 for TTL inputs, $V_{th} = 1.4 \text{ V} \pm \text{supply tolerance}$

Figure 15. Schmitt-Triggered Input With Selectable Hysteresis.

I/O TYPE: Schmitt-Triggered Input with Limited Hysteresis

MACRO NAME: None

SCHEMATIC:

CONFIGURATION:

IOB 1 - Output

I:

BUF:ON

IOB 2 - Input

I:PAD

BUF:

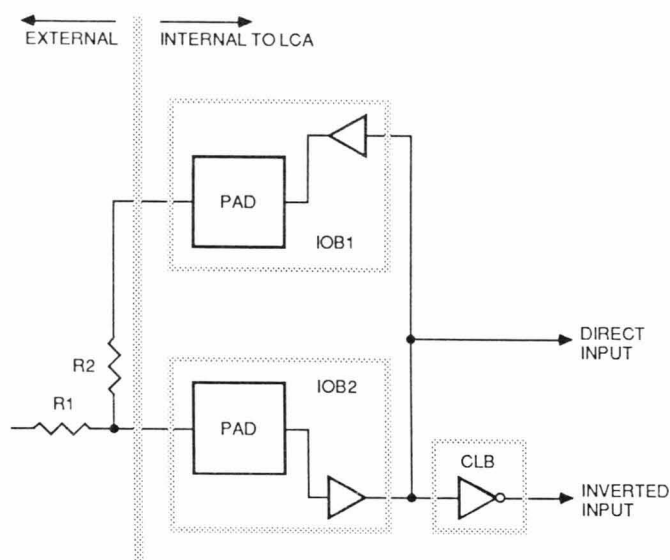
COMMENTS: Hysteresis values are limited

$$V_l = [(R_1 + R_2) / R_2] V_{th} V_{oh}(R_1/R_2)$$

$$V_h = [(R_1 + R_2) / R_2] V_{th}$$

[5]

[6]



0010022 16

Figure 16. Schmitt-Triggered Input With Limited Hysteresis.

OSCILLATORS USING IOBs

Overview

General purpose oscillators can be built using two Input/Output Blocks (IOBs) and a Configurable Logic Block (CLB). The general theory of operation is similar to that described for Schmitt triggers. For the oscillator described below, an oscillating signal is generated by the charging and discharging of two capacitors. The circuit is shown in Figure 18. Capacitor C2 charges to a voltage threshold (on Set) to set a latch. Once the voltage across C2 exceeds the threshold, the SET line causes the "Q" line to go high and starts discharging C2 by driving the IOB called CQL. After crossing the threshold, the RESET line, which has been held low, is allowed to rise as capacitor C1 charges. Once capacitor C1 charges to its threshold, the "Q" output is reset and forced low. Capacitor C1 is now discharged by the IOB named CQ, and capacitor C2 begins charging again. This process repeats, creating a low-frequency resistor-capacitor oscillator.

The designer should consider the routing delay of the three-state control lines within IOBs marked as CQ and CQL in Figure 18. The time period of the oscillator depends on each capacitor being completely

discharged during the opposite timing phase. In addition, the timing depends on both capacitors beginning their charge near ground. A routing delay difference between the output (O) of an IOB and the three-state control (T) may prevent the capacitors from completely discharging.

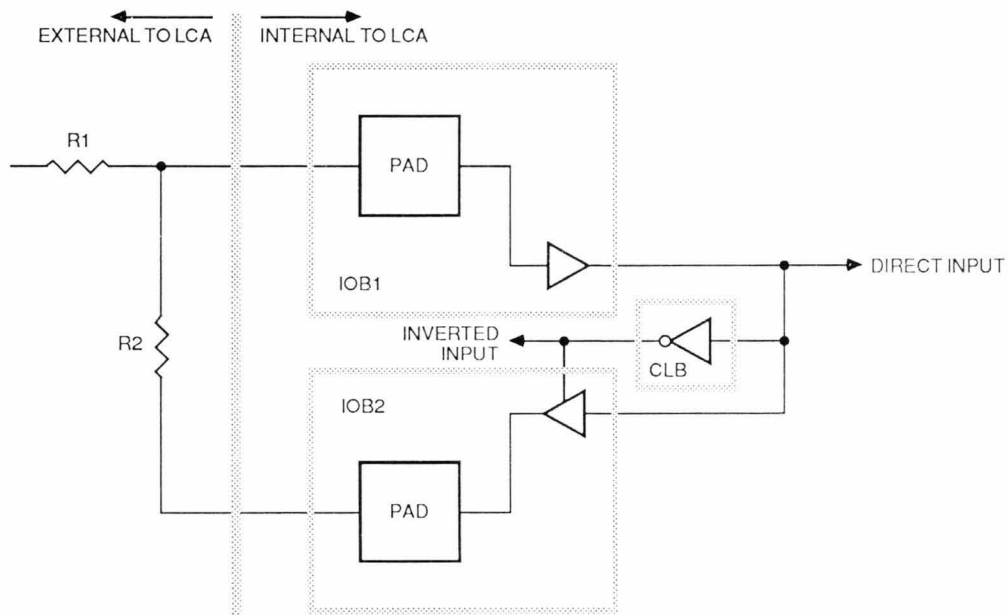
Any number of these low-frequency oscillators can be used in a design. In most designs, however, only one or two are required. If the oscillator output is used throughout the design to clock the registers within the CLBs, place the oscillator near one of the clock buffers and use the clock buffer. Figure 18 shows a low-frequency oscillator built near the main clock buffer in the upper left-hand corner of the die. A similar low-frequency oscillator may drive the auxiliary clock buffer located in the lower right-hand corner of the die.

The designer should be aware that the low-frequency oscillator circuit will cause an error when using the timing calculator to examine the oscillator. The timing calculator in the XACT Development System detects combinatorial loop conditions and flags them as errors. The oscillator circuit depends on combinatorial loopback for operation and will cause an error message. Such error messages can be safely ignored if they are detected only in the oscillator circuit.

I/O TYPE: Unidirectional Schmitt-Triggered Input

MACRO NAME: None

SCHEMATIC:



$$V_h = V_{th}$$

$$V_l = V_{th} [(R1+R2)/R2] - V_{oh}(R1/R2)$$

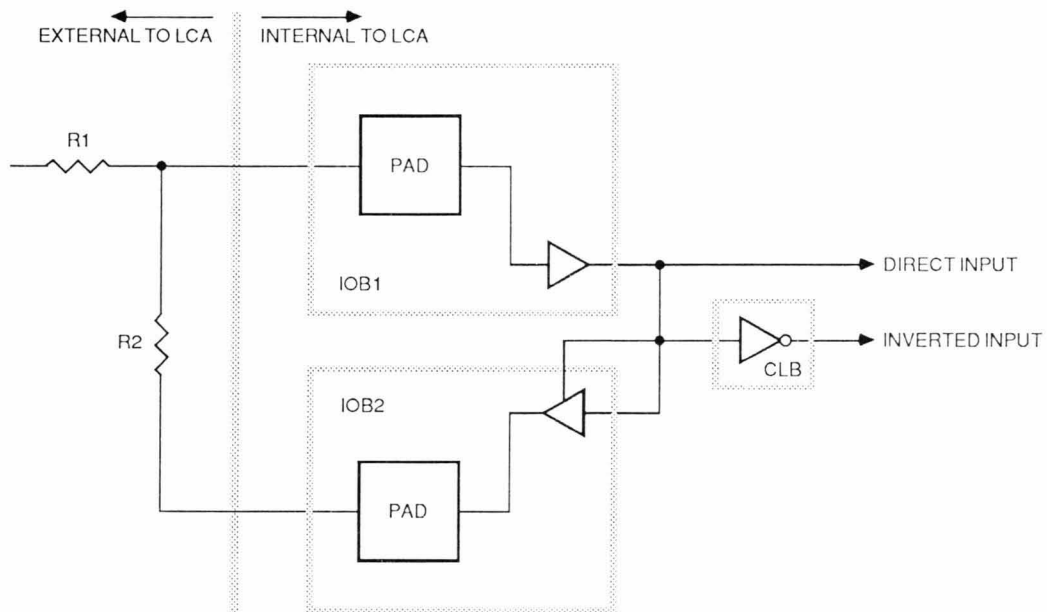
[7]

[8]

0010022 17a

Figure 17a. Unidirectional Schmitt-Triggered Input HIGH Going LOW

SCHEMATIC:



$$V_h = V_{th} [(R1+R2)/R2] - V_{ol}(R1/R2)$$

$$V_l = V_{th}$$

[9]

[10]

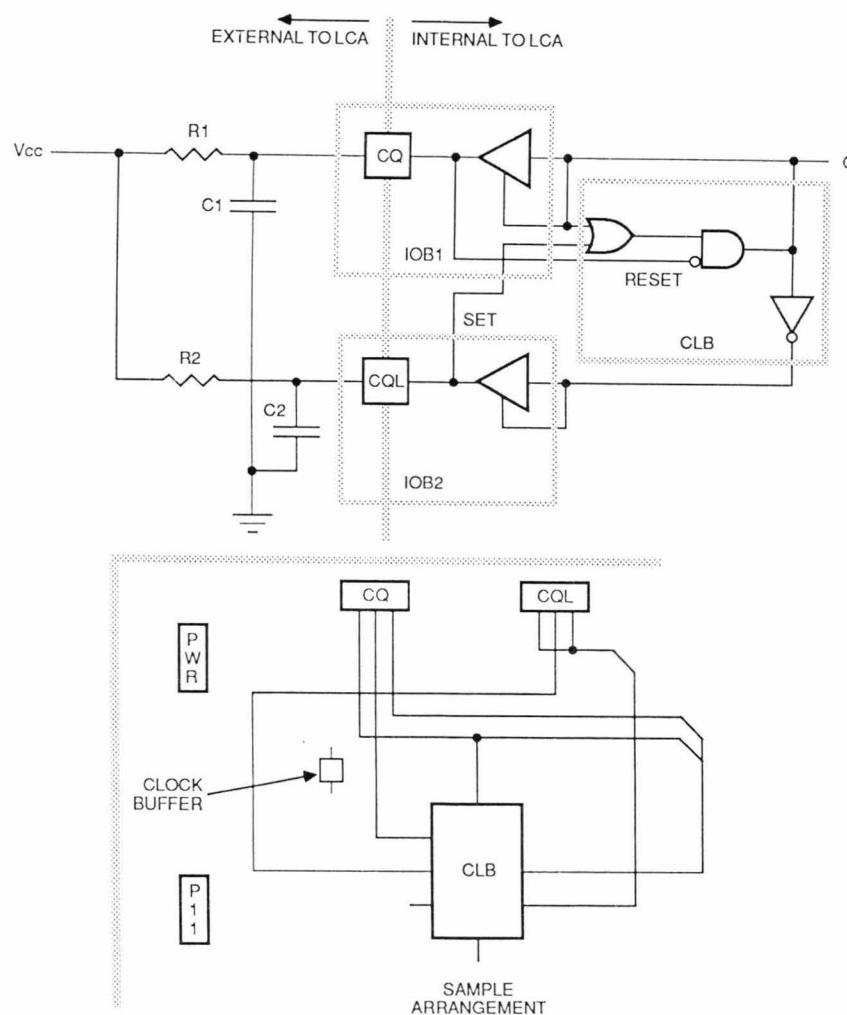
0010022 17b

Figure 17b. Unidirectional Schmitt-Triggered Input LOW Going HIGH

I/O TYPE: Low-Frequency Resistor-Capacitor Oscillator

MACRO NAME: GOSC

SCHEMATIC:



CONFIGURATION:

IOB1

I:ON

BUF:TRI

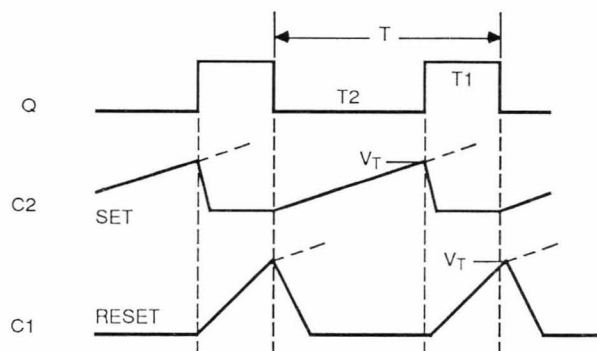
IOB2

I:ON

BUF:TRI

0010022 18

Figure 18. Low-Frequency Resistor-Capacitor Oscillator



$$T \text{ (time period)} = T_1 + T_2 = N ((R_1 \times C_1) + (R_2 \times C_2)) \quad [11]$$

where N = approximately 0.35 for TTL threshold
= approximately 0.75 for CMOS threshold

Assumptions: Each capacitor is discharged during opposite timing phase. Capacitors begin charging from GROUND. Effect of three-state routing delay is assumed minimal.

0010022 19

Figure 19. Low-Frequency Resistor-Capacitor Oscillator Timing Diagram

ON-CHIP CRYSTAL OSCILLATOR

Basic Description

Two special user-defined I/O Blocks can be configured to interface directly to the on-chip crystal oscillator located in the lower right-hand corner of the die. The crystal oscillator is associated with the auxiliary clock buffer located near the oscillator. When the interconnect is selected to drive the auxiliary clock buffer two special pins interface directly to the internal high-speed inverting amplifier to form the oscillator. Externally, these pins should be attached to crystal oscillator components as shown in Figure 20. The best way to configure the crystal oscillator is through the MACRO named GXTL.

Even before device configuration is complete, the on-chip oscillator begins operation so that the circuitry can stabilize. The actual internal connection of the oscillator to other circuitry on the chip is delayed until completion of device configuration.

Theory of Operation

The feedback resistor (R1) from output to input biases the amplifier at threshold and should be as large a value

as practical up to 4 M Ω . The inversion and delay of the amplifier, together with the R-C networks and crystal, produce a 360 degree phase shift, forming a Pierce oscillator. The series resistor (R2) may be included to add to the amplifier output impedance when needed for phase shift control, crystal resistance matching, and to limit the amplifier input swing to control clipping at large amplitudes.

Excess feedback voltage may be adjusted by the ratio of capacitors C2/C1. The amplifier is designed for use in the range from 1 MHz up to one-half the specified CLB toggle frequency. Using the oscillator at frequencies below 1 MHz requires individual characterization with respect to a series resistance. Operation at frequencies above 20 MHz is more involved since it generally requires that the crystal operate in a third overtone mode in which the fundamental frequency must be suppressed by the R-C networks.

REGISTERS IN IOBs

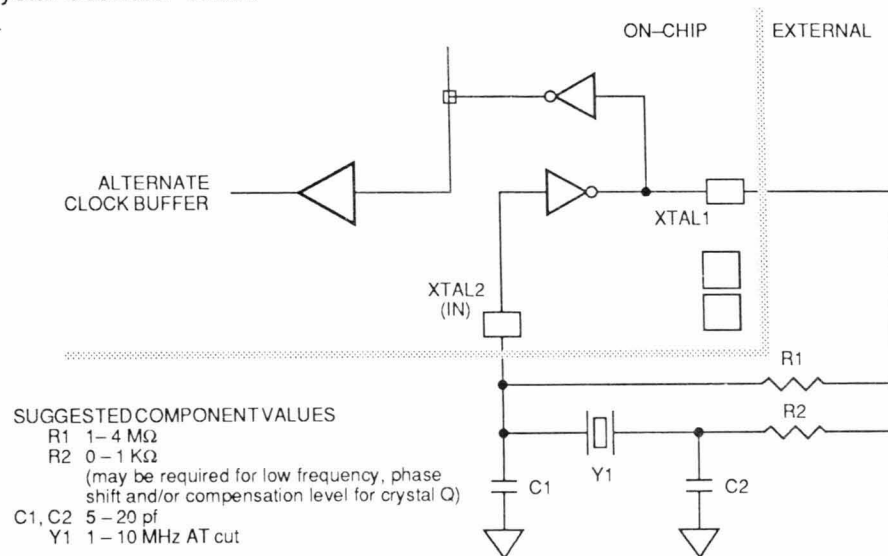
Overview

The previous examples describe the use of IOBs for conventional applications. All involved using either the

I/O TYPE: On-Chip Crystal Oscillator Circuit

MACRO NAME: GXTL

SCHEMATIC:



	XTAL1	XTAL2
48 DIP	33	30
68 PLCC	46	43
68 PGA	J10	L10
84 PLCC	56	53
84 PGA	K11	L11

0010003 10

Figure 20. On-Chip Oscillator Circuit

input, or the output, or both. If an IOB is not required for input or output, the storage element within each IOB can be used to create registers and various types of counters. All of the following designs involve using the output buffer (BUF:ON) fed back into the input register (I:Q). This configuration, which is used with slight modifications in each of the following examples, is shown in Figure 21. The pads of the IOBs involved are not typically connected to anything externally, although they may be if desired.

IOB-based Register Delays

The delays incurred through an IOB-based register depend on the sum of two parameters—the delay through the output buffer and the delay back through the input buffer to the register. While these values are defined in the data sheet for an output load of 50 pF, their values change only slightly for no output capacitance. The delay into an IOB-based register is:

$$T(\text{IOB-reg}) = T_{op} + T_{pl} \quad [12]$$

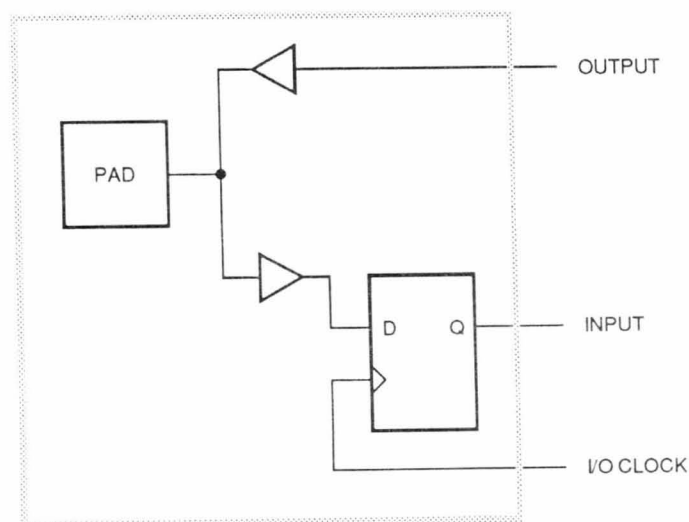
where:

T_{op} = Output to Pad output
 T_{pl} = Pad input set up to I/O clock (min)

I/O TYPE: Pad with Input Storage (IOB-based register)

MACRO NAME: PREG

SCHEMATIC:



0010022 21

CONFIGURATION:

I:Q
 BUF:ON

Figure 21. Pad With Input Storage (IOB-Based Register)

Wide Storage Registers

Wide storage registers can be built from the basic structure described in Figure 21. For example, Figure 22 shows the construction of a n-bit storage register built from IOBs. Wide storage registers are ideal for IOBs, since the I/O clock feeding an IOB is common to all IOBs along each edge of the die.

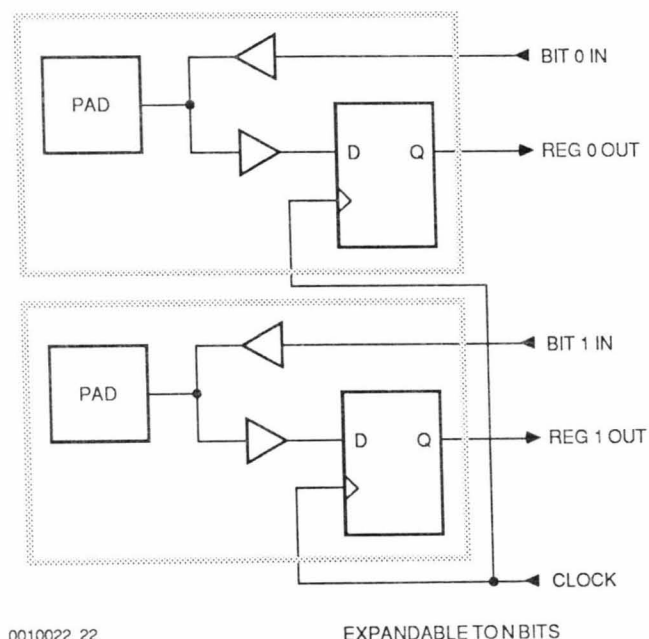
Read/Write Registers

Another variation of the basic IOB-based register is a simple read/write register. This structure allows data to be written into registers within the LCA from an external device and also read back. Figure 23 shows the structure of a read/write register. In this example, the

I/O TYPE: N-bit Storage Register (IOB-based register)

MACRO NAME: None

SCHEMATIC:



0010022 22

CONFIGURATION:

All IOBs
 I:Q
 BUF:ON

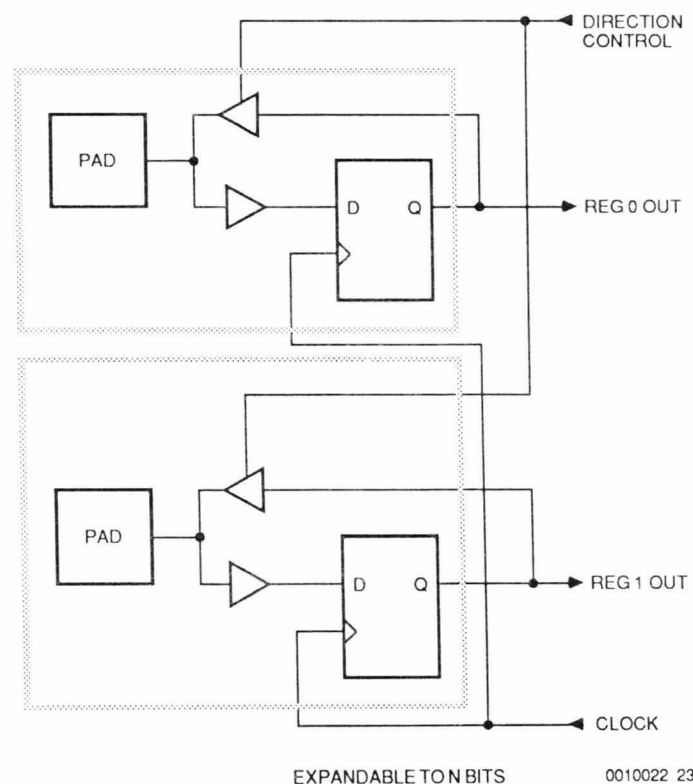
COMMENTS: The I/O clock into each IOB is common to all IOBs along each edge of the die. For best resource utilization, group the storage elements along one edge of the die.

Figure 22. N-Bit Storage Register (IOB-Based Register)

input (I) and output (O) stubs of the IOB are connected together. The three-state control line (T) controls the direction of data flow (T = LOW for a read operation by the external device, T = HIGH for a write operation to the LCA). Typically, the read/write control line (three-state control) also originates external to the LCA, and would come in through an additional I/O block.

The designer should be aware that the input register data can be read from the LCA from the read/write

I/O TYPE: N-bit Read/Write Storage Register
MACRO NAME: None
SCHEMATIC:



CONFIGURATION:
 All IOBs
 I:Q
 BUF:TRI

COMMENTS:

The I/O clock into each IOB is common to all IOBs along each edge of the die. For best resource utilization, group the storage elements along one edge of the die.

The three-state control line controls the direction of data flow (T = LOW for read, T = HIGH for write). This control line will typically originate off chip and come in through an additional IOB.

Figure 23. N-Bit Read/Write Storage Register

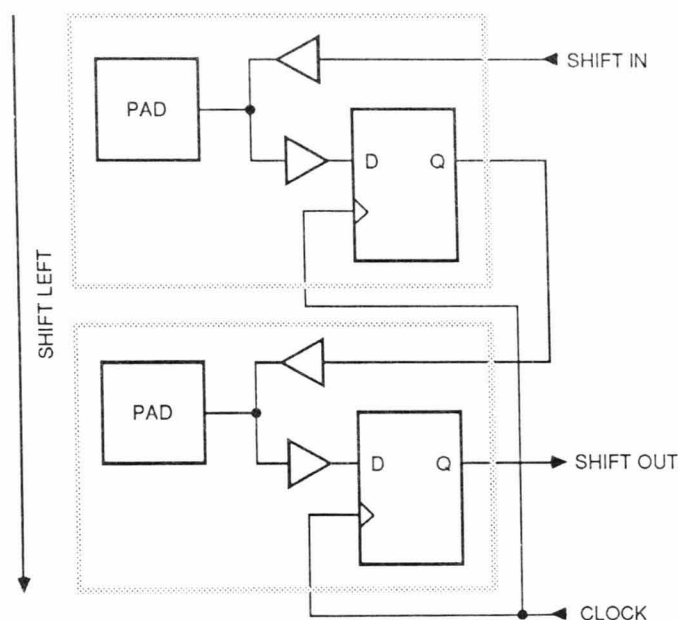
register but data cannot be written to the LCA. Writing the register from inside the LCA would require that two network sources be active, which is not allowed.

Shift Registers

Shift registers are also easily constructed with IOBs by feeding the input (I) of one IOB to the output (O) of the next IOB. The figures below describe two shift registers—one that shifts to the left as shown in Figure 24, and one that shifts to the right as shown in Figure 25. The shift direction of the register depends on how the inputs and outputs of the IOBs are connected.

Since the I/O clock line of an IOB is common to all IOBs along each edge of the die, the register is implemented with IOBs along one edge of the die.

I/O TYPE: Shift Left Register (IOB-based register)
MACRO NAME: None
SCHEMATIC:



CONFIGURATION:
 All IOBs
 I:Q
 BUF:ON

COMMENTS: Notice that the routing of the input (I) of a given IOB goes to the output (O) of the IOB on the left (shift left).

Figure 24. Shift Left Register (IOB-Based Register)

Johnson Counters

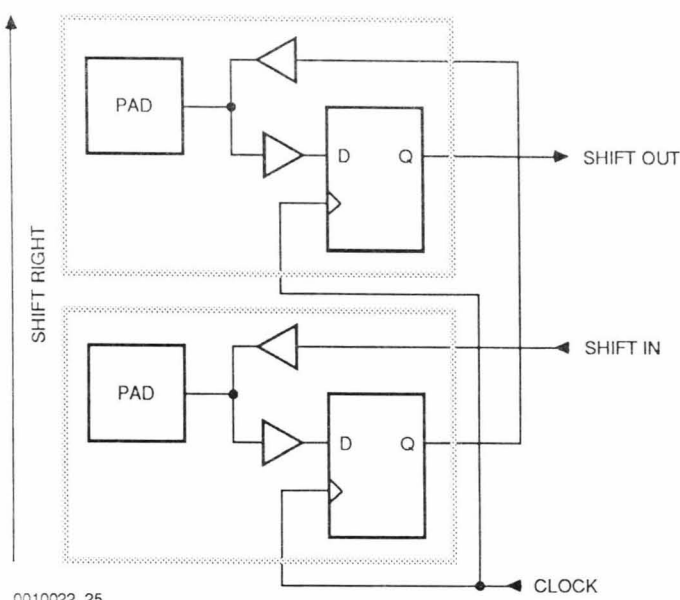
An n-bit Johnson counter will count to $2n$ states as opposed to standard binary counters which count to 2^n possible states. Johnson counters have a variety of possible uses in a digital design, including low modulo counters and glitch-free decoders.

In IOB implementations, Johnson counters can be thought of as special shift registers. Only one bit changes during a state transition, as shown in Table 1 for a three-bit Johnson counter.

000
100
110
111
011
001
000

Table 1. Transitions of a Three-Bit Johnson Counter

I/O TYPE: Shift Right Register (IOB-based register)
MACRO NAME: None
SCHEMATIC:



CONFIGURATION:

All IOBs
I:Q
BUF:ON

COMMENTS: Notice that the routing of the input (I) of a given IOB goes to the output (O) of the IOB on the right (shift right).

Figure 25. Shift Right Register (IOB-Based Register)

A Johnson counter built from unused IOBs requires at least one Configurable Logic Block (CLB) to perform an inversion. The Johnson counter is automatically reset to an all zeroes state upon configuration or on a RESET pulse.

The application note *Counter Examples* contains more information on using Johnson counters.

Glitchless Johnson Decoder

A glitch-free decoder can be built using IOBs and CLBs. The decoder will be glitch-free since only one bit changes during a state transition. An n-bit Johnson counter decoder can decode any one of the $2n$ possible states or any number of contiguous states by ANDing just two of the appropriate counter bits. Counters of various modulo and duty-cycle can be extracted as well. For example, Figure 27 shows the schematic implementation of a Johnson counter decoder with various two-input decode states.

Linear Feedback Shift Registers

Linear Feedback Shift Registers (LFSRs) are yet another modification of a simple shift register. An LFSR consists of a shift register with feedback of appropriate bits back to the first bit. An LFSR requires some logic function in the feedback path, usually a XOR (exclusive-OR) function.

LFSRs have numerous applications. One example is described in the application note *A UART Design Example*. In the UART example, LFSRs are used to implement the encryption and decryption functions. The application note *Counter Examples* contains more information on using Linear Feedback Shift Registers in general applications. This application note addresses using LFSRs in IOBs.

Figure 28 shows the schematic for a three-bit LFSR which implements a modulo 5 (divide by five) counter. An n-bit LFSR counter can produce a pseudorandom sequence of up to $2^n - 1$ unique states. By adding logic to the feedback path, the LFSR counter can be forced to skip any number of states (from one to $2^n - 1$). By forcing the counter to skip m states, a LFSR counter can implement any modulo as described in Equation 13.

$$\text{MODULO} = (2^n - 1) - m \quad [13]$$

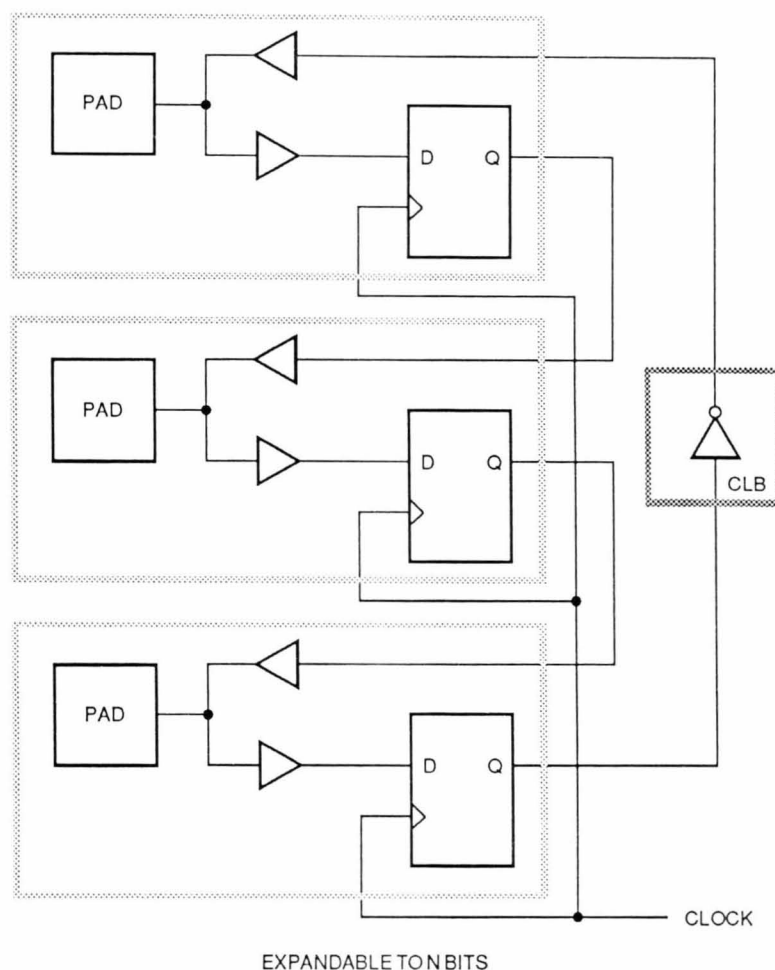
where n = number of shift-register bits
m = number of "skipped" states

Figure 29 shows the counting sequence for a three-bit LFSR counter with exclusive-NOR (XNOR). All of the possible "skip" paths are indicated. The "stuck" state is also shown.

I/O TYPE: N-bit Johnson Counter

MACRO NAME: None

SCHEMATIC:



0010022 26

CONFIGURATION:

All IOBs

I:Q

BUF:ON

Figure 26. N-Bit Johnson Counter (IOB-Based)

In the counting sequence, note that there are two counter states where only the first bits differ (for example, locate the states 101 and 001). By forcing the feedback logic to invert the sense of the feedback into the first bit, the counter can be forced to “skip” all of the states between the two indicated values. This can be accomplished by decoding (ANDing) the state just previous to the state to be skipped. Again using the modulo 5 counter as an example, locate the initial value that will allow the counter to skip two states (i.e. 101). By decoding the state 011 (the state just prior to the initial

skip state, 101), the sense of the feedback into the first is inverted. The counter skips from state 101 to state 001 implementing a modulo 5 counter. Using this method and the proper feedback into the register, a counter of any modulo from one to $2^n - 1$ can be built.

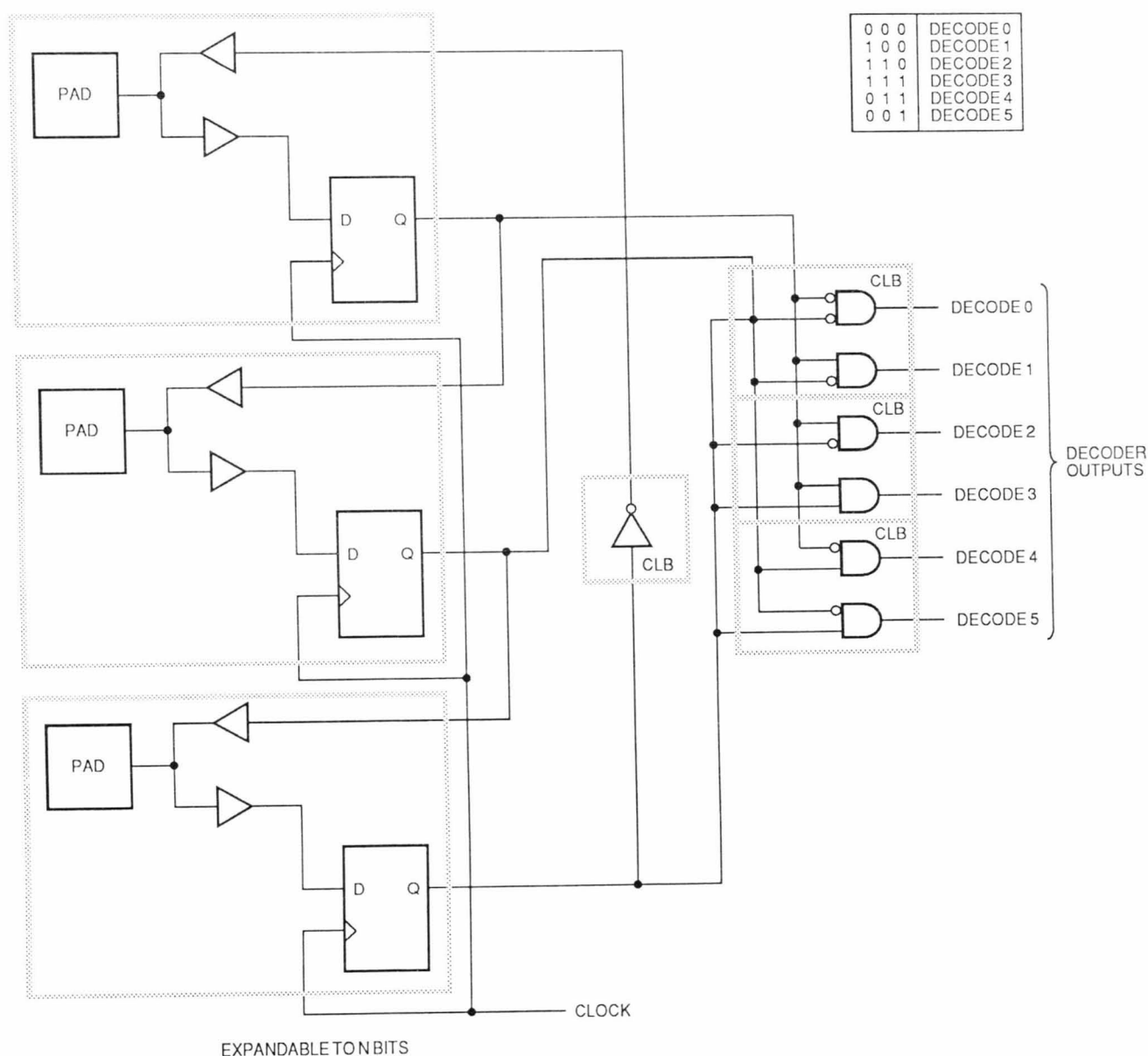
Upon configuration or upon an externally driven $\overline{\text{RESET}}$ signal, all of the storage elements used in the LFSR counter will be reset to zero.

The designer should be careful to avoid the “stuck”

state. The “stuck” state is the state missing from the 2^n-1 counting sequence (if the “stuck” state were included, the LFSR counter could have 2^n possible states). This state occurs when the feedback path forces the counter into an ever-repeating single state. As a simple example, assume that a LFSR counter were built with a two-input exclusive-OR feedback path as shown in Figure 30. Upon configuration or external RESET, the counter would begin operation in the all zeroes state (000) and would be “stuck” in that state due to the type of feedback used.

An interesting thing occurs when all but the last bit of the “stuck” state are decoded (ANDed together) and included in the feedback path. Instead of counting over a possible range of 2^n-1 states, the extra decoding causes the LFSR counter to count to all 2^n states as shown in Figure 31.

Longer LFSR counters with higher possible modulus and more complex feedback mechanisms can be built but their discussion is well beyond the scope of this application note. However, Table 2 presents some of



0010022 27

Figure 27. Johnson Counter Decoder. Any state of a Johnson counter can be decoded, glitch-free, with only a two-input logic function.

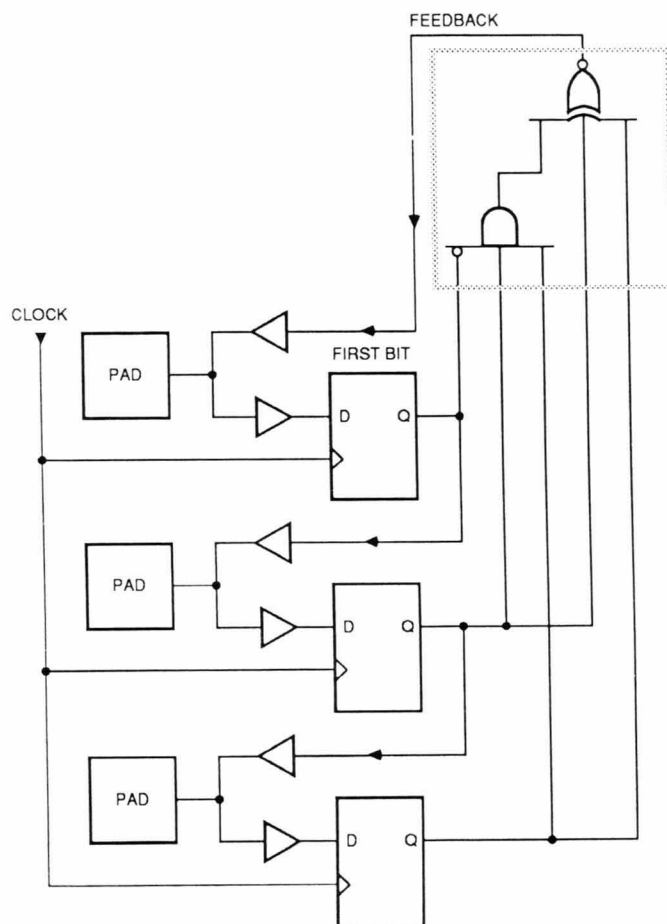
the possible feedback combinations for LFSR counters of three bits to ten bits.

$(2^n - 1)$ Modulo	7	15	31	63	127	255	511	1023
Feed- back Options into Bit 1	1,3 2,3	1,4 3,4	2,5 3,5	1,6 5,6	1,7 3,7 4,7 6,7	1,2,7,8	4,9 5,9	3,10 7,10

Table 2. I/O Block

Current Drive			
	4 mA	6mA	8mA
Voh	3.86 V	3.54 V	3.22 V
Vol	0.32 V	0.48 V	0.64 V

Table 3. Output Current and Output Voltage Levels for an IOB



00100252 28

Figure 28. Modulo 5 LFSR Counter

Enhanced Output Source/Sink Current

Logic Cell Arrays are specified to have 4 mA worst case source and sink capabilities at $V_{ol} = 0.32$ Volts and $V_{oh} = 3.68$ Volts. Increased drive current can be obtained at the cost of decreased voltage margins. For example, Table 3 illustrates the effect on V_{ol} and V_{oh} by increasing the drive current through a single IOB.

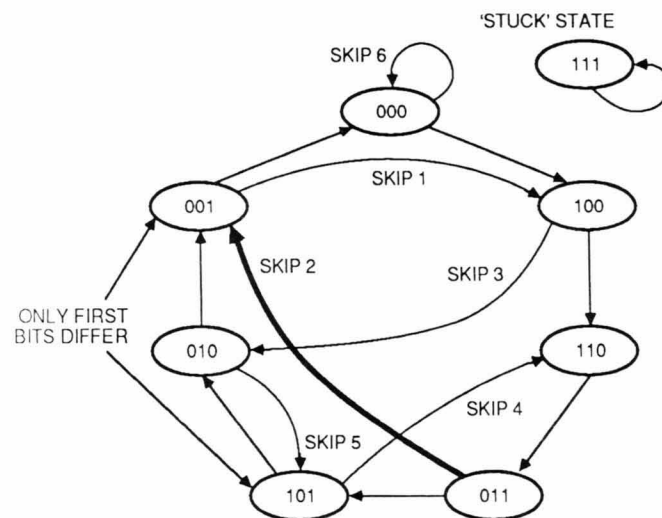
An alternative method of increasing the drive current is to parallel the output drivers of two IOBs. Paralleling two outputs will enable the IOBs to source and sink double the current (worst-case) at no reduction in voltage margins. This method is schematically diagrammed in Figure 32.

One caution to the designer, however. The designer should minimize the difference in routing delay between the two IOBs connected in parallel. Excessive delays may cause output contentions.

SUMMARY

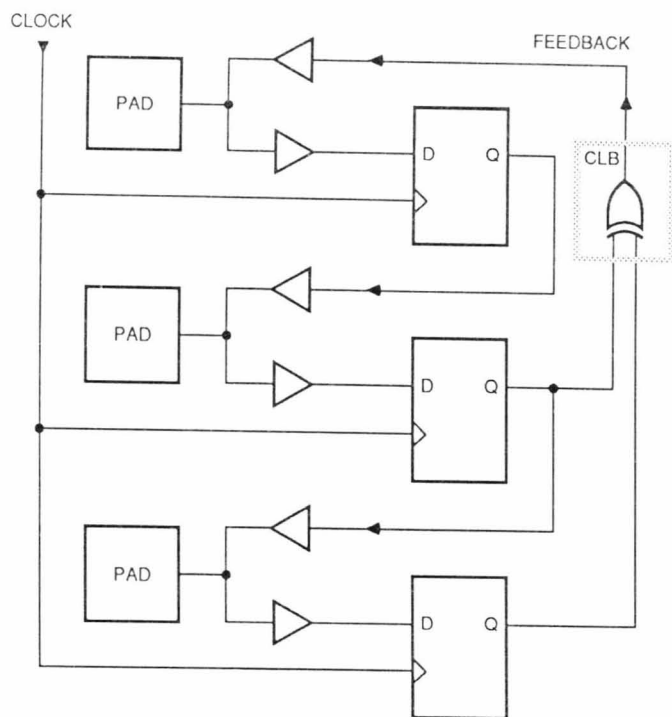
The input and output resources of a Logic Cell Array (LCA) can be used for a variety of logic structures because of the flexibility of the LCA architecture. This application note described how to implement various I/O structures including bidirectional lines, open-drain outputs, and Schmitt-triggered inputs.

Other structures such as oscillators, multiplexers, shift



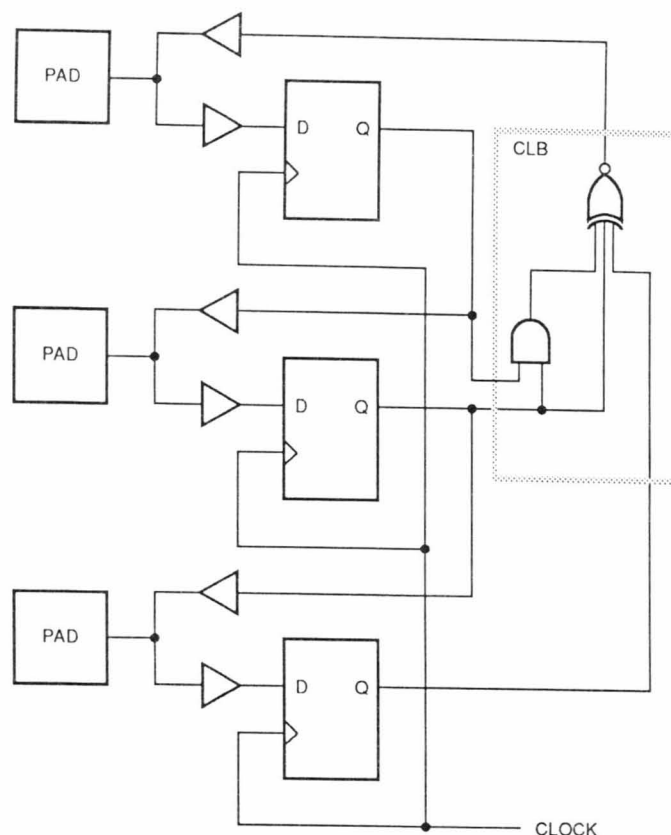
0010022 29

Figure 29. Three-bit LFSR Counting Sequence



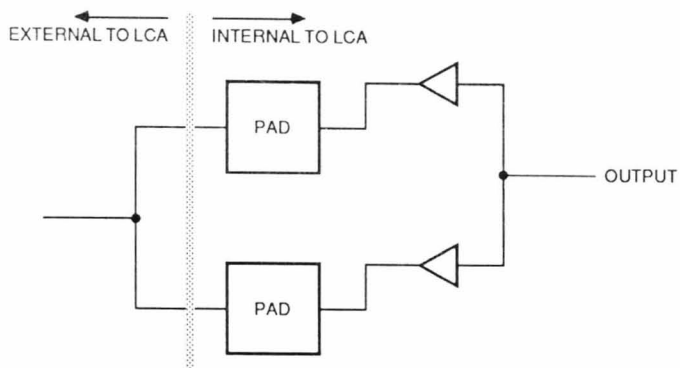
0010022 30

Figure 30. Simple LFSR With "Stuck" State. A simple LFSR counter will be "stuck" in state 000 after configuration since all registers were originally reset.



0010022 31

Figure 31. By ANDing all but the last bit of the "stuck" state and using this value in the feedback path, an LFSR can be forced to count to 2^n possible states instead of the 2^n-1 states usually associated with an LFSR counter.



0010022 32

Figure 32. Parallel Outputs Have Increased Drive Capability

registers, Johnson counters, decoders, and linear feedback shift registers were also described. These resources can only be effectively implemented in flexible, array-type architectures such as found in gate arrays and the Xilinx Logic Cell Array.

TECHNICAL SOURCES

¹*Schmitt Trigger Using PLS153 and PLS159*, Signetics Programmable Logic Data Manual, 1986. pp. 9-110 to 9-119.