# Low Power Design with CoolRunner-II CPLDs

**XILINX**®

XAPP377 (v1.0) May 8, 2002

## Summary

CoolRunner™-II RealDigital CPLDs are the only CPLD to combine both high performance and low power to form the next generation CPLD. This application note describes the design methodologies that can be employed to obtain the lowest power possible using the CoolRunner-II CPLD by utilizing its unique power saving features.

## Introduction

CoolRunner-II CPLDs continue to employ the features of Fast Zero Power™ (FZP) technology as found in the earlier CoolRunner CPLD generations. But due to unique Xilinx design techniques, smaller geometries, and state of the art process technology, FZP technology has further advanced CoolRunner-II CPLDs as the low power standard. Other CPLD manufacturers have attempted to reproduce the FZP technology of the CoolRunner CPLDs, but have not been able to meet the CoolRunner benchmark. For the first time in the CPLD industry, CoolRunner-II devices deliver both true high performance and low power at the same time, along with the lowest standby current in the industry without the use of power down modes. In addition, these devices offer unique power saving features such as CoolCLOCK, DataGATE, and DualEDGE flip-flops.

Traditional CPLDs use sense amp type product terms to provide fast propagation delay times. Sense amp product terms are biased in a manner to detect a small change in voltage levels on the word line which indicates a change in the logic state of the product term. The transistor biasing constantly draws current, even at standby. With this in mind, these types of CPLDs cannot provide a low power solution, as can CoolRunner-II CPLDs. As sense amp type device sizes become larger in macrocell count, power grows significantly since there are many more product terms to consume power.

As process technology shrinks, sense amp type CPLDs will inherently consume more power to maintain performance. Smaller process technology demands lower power supply voltages thereby reducing the gain of the sense amp. Further, product term transistors leak more with the smaller geometries. To maintain performance, a sense amp CPLD will need to be designed such that its biasing compensates for the leakage and boosts gain to detect the smaller voltage swing of the word line. Higher biasing will cause more current to flow through the bias network, thereby increasing total power consumption. Other CPLD manufacturers that use sense amp based technology will inevitably go through a learning process to re-design their current products to compensate for the effect of ever shrinking process technology.

CMOS product terms, as used in the FZP technology, inherently consume less power when not switching states. Since CMOS logic exhibits low standby current, CoolRunner-II CPLDs use this technology to reap the benefits of low power. Additionally, CMOS technology benefits from smaller geometries where the device consumes less power and becomes faster.

## Power Saving Features

New architectural features have been added to the CoolRunner-II product line to enhance the power saving capabilities of the FZP technology. This section describes these features and how they can be used to save power.

All new features described below are available with the XC2C128 and larger devices. The smaller devices, XC2C32 and XC2C64, do not include the DataGATE, Clock Divider, and CoolCLOCK features.

## DataGATE

Many times devices are connected to a data bus which are not being addressed by the master device. When a CPLD is in this situation, it will use more power than necessary since the data on the bus is not useful to the CPLD, but the data lines continue to toggle, which subsequently toggles the internal logic of the CPLD. Any logic that changes state within the CPLD will consume power. Therefore, it follows that disconnecting the CPLD from the data bus when the device is not addressed conserves power.

DataGATE solves this issue by disconnecting external signal activity from the internal logic, consequently reducing power consumption of the device. This is achieved by using a unique, software enabled, CoolRunner-II pass gate at the I/O pin (when configured as an input) which is controlled by the DataGATE global net. This control net originates from a specific pin/macrocell and can be driven by either an external signal or an internally developed signal using logic elements. When the pass gate on the input pin is disabled by the DataGATE control net, an internal latch drives the CPLD logic network maintaining the same logic level that was present on the input pin just prior to asserting the DataGATE control net. This preserves the current logic state internal to the CPLD while external data changes states.

For example, a CoolRunner-II CPLD that shares a data bus with other devices will most likely have its own address and typically will not be addressed continuously. Two options exist to disconnect the data bus from the CoolRunner-II CPLD when not addressed. First, if the device is addressed using a chip select signal, this signal can be assigned to the DataGATE control pin and used to isolate the inputs from the data bus. Second, if using an address bus, the address of the device can be decoded internally to the CPLD which can then be used to enable, via DataGATE, the data bus inputs to receive data when addressed. When the CPLD is not addressed, DataGATE would disconnect the external data bus signals using address decoding internally to the CPLD. In this case, the result of the internal decoding is routed to the DataGATE pin to disconnect the toggling bus.

DataGATE can be configured to affect any or all I/O pins, with the exception of JTAG pins and the DataGATE pin itself. The above example discusses DataGATE configured to affect a few macrocell I/O pins. It may be beneficial in other applications to disconnect the system clock or clocks from the CPLD. The two elements that consume the most power are output buffers and the clock tree. If the CPLD is not needed for a specific amount of time, the DataGATE feature could be used to gate the clock. Doing so will dramatically reduce power consumption. However, caution must be exercised when gating a clock since undesired logic transitions may occur.

There are other cases where the unique CoolRunner-II DataGATE feature is useful to reduce power. Also, DataGATE is flexible such that the entire device or only parts of the device can be isolated from external signals, depending on the application. For example, it can be used in CoolRunner-II CPLDs to enhance board troubleshooting procedures.

## Schmitt Trigger

CoolRunner-II Schmitt trigger inputs are useful for applications that require hysteresis on the inputs. A useful application might be where noise is an issue on specific pins. Hysteresis provides added noise immunity. Another application could implement an oscillator circuit which is constructed external to the CPLD, but CPLD logic is used to implement portions of the oscillator circuit.

To ensure the lowest power consumption in the CoolRunner-II CPLDs, disable the Schmitt trigger inputs since these types of buffers consume more power than the regular input buffer. This statement assumes that input signals consist of low noise, fast rise time edges. It is important to design and operate the system in a low noise environment when Schmitt triggers are disabled to prevent inadvertent transitions on the CPLD inputs.

Conversely, when the input signals consist of slow rise time edges, it is likely that enabling Schmitt triggers will save power consumption. Since slowly rising signals linger in the intermediate logic state for a greater amount of time, normal input buffers will consume more power during this transition. Schmitt triggers avoid this situation by introducing hysteresis.

## Clock Divider

Global clock networks tend to be the largest power consuming elements in CPLDs. Any effort to reduce the frequency of the global clock network greatly benefits the system with respect to power consumption. Therefore, CoolRunner-II devices have been designed to include a clock divider network on global clock, GCK2. Without introducing additional clock delays, the clock divider has the capability of dividing the system clock by even integers ranging from 2 to 16, as shown in Figure 1.
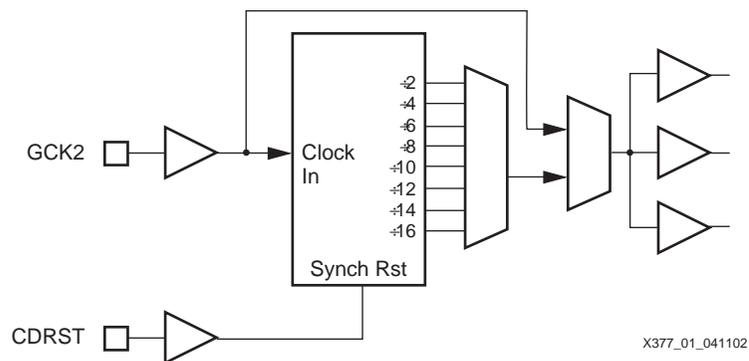


*Figure 1:* **Clock Division Circuitry for GCK2**

Some systems use state machines, for example, that do not require the full speed of the external system clock. A clock divider is the perfect tool to reduce system power in this case. The clock divider provides an excellent alternative to adding a user defined clock divider built from logic, which would waste logic otherwise usable for more features in the design. A lower frequency on the global clock network, provided by the clock divider, will reduce the power consumed by the CoolRunner-II CPLD.

Generally speaking, designing with the slowest system clock possible will reduce power consumption. To this end, the clock divider provided in the CoolRunner-II architecture will greatly assist the designer.

## DualEDGE Registers

By utilizing both edges of the clock signal, the macrocell can do twice the work when configured as a DualEDGE flip-flop. Figure 2 displays the macrocell configured as a DualEDGE flip-flop. A system without the aid of the DualEDGE flip-flop would need to provide a clock at twice the frequency to obtain the same work output at the macrocell. Since the macrocells with DualEDGE flip-flops operate on both the rising and falling edges of the clock, the clock network

is used more efficiently. Consequently, power consumption is reduced when the global clock net is operating at a lower frequency.
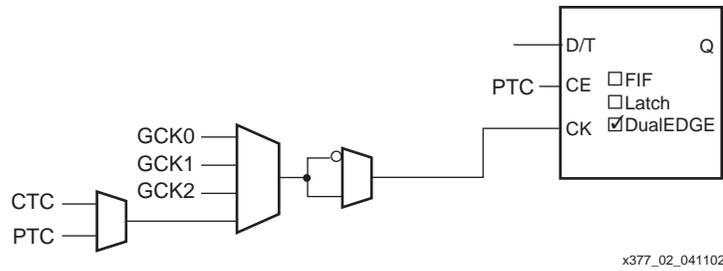


*Figure 2:* **Macrocell Clock Chain with DualEDGE Option Shown**

DualEDGE flip-flops further enhance the functional possibilities of the clock divider and therefore improve power savings. The global clock can be effectively divided by odd integers of 3, 5, and 7 if used with the DualEDGE flip-flop. For example, if a divide by 3 clock is desired for the design, the Clock Divider can be set to a divide by 6 and then effectively doubled by the DualEDGE flip-flop resulting in a divide by 3 characteristic. This is important to note since, again, lower clock frequencies always save power. The DualEDGE flip-flop effectively adds more functionality to the clock divider network.

Perhaps the design contains two state machines. One state machine can efficiently operate at 1/4th the system clock frequency, yet the other state machine can much more efficiently operate at 1/8th the system clock frequency. DualEDGE flip-flops can be employed in conjunction with the clock divider to obtain such a scenario. The state machine running at 1/8th the system clock frequency can simply use the clock divider configured as divide by 8. Enabling the DualEDGE flip-flop on macrocells assigned to the other state machine and assigning the divide by 8 clock divider to those macrocells as well will yield an effective clock frequency that is 1/4th the system clock frequency. This means that the single clock divider can obtain virtual dual functionality of a divide by 8 and a divide by 4 counter. Notice that both clocks are synchronized with each other and the system clock so that the two state machines operate concurrently.

Summarizing the previous discussion, when utilizing the CoolRunner-II clock divider and/or the DualEDGE flip-flops, it is possible to obtain clock divisors of 2, 3, 4, 5, 6, 7, 8, 10, 12, 14 and 16. Additionally, when a group of macrocells use the clock divider and some of those macrocells use DualEDGE flip-flops, the clock divider network can effectively deliver two clock frequencies of divisors 1-2 (using CoolCLOCK described later), 2-4, 3-6, 4-8, 5-10, 6-12, 7-14, and 8-16. When DualEDGE flip-flops are used with the clock divider, lower power is achieved while more efficiently operating the design.

## CoolCLOCK

CoolRunner-II CPLDs are equipped with a unique feature, CoolCLOCK, to further reduce the power consumption of the global clock network without affecting the speed of the clock. Note that CoolCLOCK does not impose additional clock delays. As explained earlier, any efforts to reduce the clock frequency of the global clock net will significantly reduce power consumption.

CoolCLOCK reduces power consumed by the global clock net by dividing the external clock frequency by 2 before it is applied to the global clock network. This clock division occurs early in the clock tree near the clock input buffer so that the divided clock affects the majority of the clock network. Since the global clock network contains a relatively large amount of internal capacitance, a slower frequency will significantly reduce power consumed by this net, GCK2. This divided clock signal is then effectively doubled at the macrocell using the DualEDGE clocking feature of the flip-flops in the macrocell, shown in Figure 3. This ensures that the original clock frequency is applied to the macrocell as intended by the external system. Only macrocells that require the original clock frequency will be configured to utilize the DualEDGE

flip-flop feature. Other macrocells can be configured to use the divided clock frequency to further reduce power when those macrocells don't require clocking at full speed.
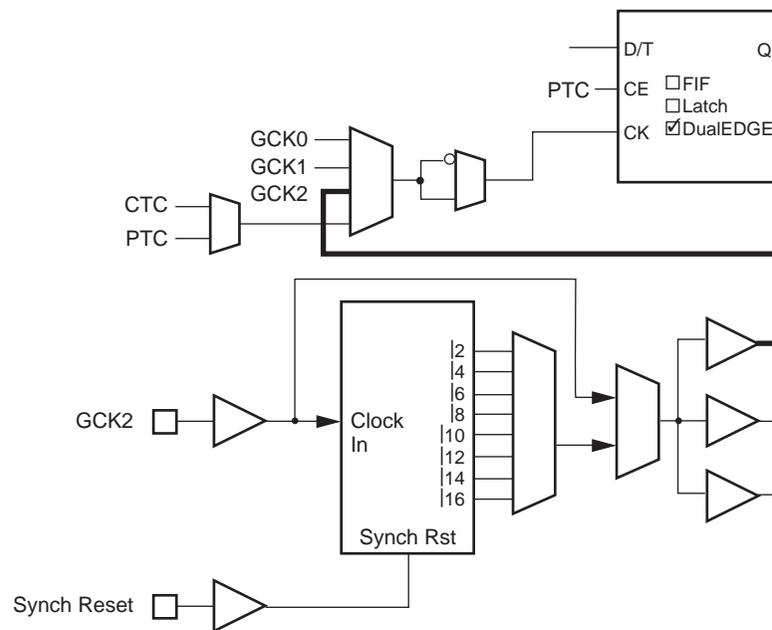


X377_03_041102

*Figure 3:* **CoolCLOCK Created by Cascading Clock Divider and DualEDGE Option**

## Weak Pull-up and Bus Hold

All CoolRunner-II CPLDs include I/O termination options to reduce power consumption of the I/O due to externally 3-stated busses. The I/O termination circuitry can be configured in three ways: weak pull-up, bus hold, and no termination. Pull-up and bus hold are selected on a global basis and are mutually exclusive. Subsequently, the use or absence of the selected termination is specified on a per pin basis. Weak pull-up connects a high-impedance resistive load onto the I/O pin to prevent a floating situation on the pin. Bus hold is essentially a full latch on the I/O pin which drives the last state present on the I/O, either High or Low, prior to the bus going to the high impedance state. Bus hold is referred to in the software as "Keeper".

Floating inputs, an input which is not driven to a High or Low logic state, can use excessive power since the voltage on the gate of the input buffer may wander to a voltage level between standard logic levels. In this case, the input buffer is driven into the linear region where the P and N channel transistors are both switched on. Therefore, to avoid this situation, I/Os can be configured to use internal pull-ups or bus hold circuitry. I/Os configured as inputs or bidirectional should utilize this feature if it is known that the bus to which the I/O is attached will float at some point in its regular operation.

There are some cases where weak pull-up is undesirable. If the bus is pulled down the majority of time, current will flow through the weak pull-up to ground via the buffer, pulling the bus Low. A design such as this would benefit by using the bus hold circuitry.

A rule of thumb for any CMOS device is to not allow inputs to float. These two features of CoolRunner-II CPLDs, weak pull-up and bus hold, will minimize the chance of consuming excessive power on input pins.

I/O termination should be considered for each application and each I/O to determine the best combination to reduce power consumption. Again, avoid floating inputs whenever possible.

### Slew Rate

This is a feature of the I/O structure that regulates the rate at which the output buffer changes states. There are two modes: FAST and SLOW. Designers concerned with reducing reflections on circuit board traces, minimizing RFI or minimizing EMI should specify a SLOW slew rate. Most design engineers considering low power designs will usually be designing slower speed systems and therefore will not be as concerned with reflections. In a case such as this, the system will benefit from a lower power perspective when slew rate is specified to be FAST.

Although an I/O is configured as an output, the input buffer is still connected to the pin and therefore senses changes in voltage on that pin. If the output is configured with a SLOW slew rate, the output voltage switches states less quickly, thereby lingering longer between standard logic levels. The input buffer will therefore be driven in the linear region (the input voltage between standard logic levels) for a longer period of time than if the output was configured in the FAST slew rate mode. Current will flow through the input buffer P and N channel transistors for a longer period of time resulting in higher power consumption. It is therefore recommended to configure the output with a FAST slew rate whenever reflections are not a concern.

## Power Saving Techniques

Several rules of thumb should be followed when designing any circuit with CMOS devices to reduce power consumption. A basic understanding of power consumption must be reached prior to discussing these rules of thumb. Therefore, a derivation of the current equation for CMOS devices is necessary. Once the mathematical model of current is understood, it becomes easy to follow the theory of the rules of thumb. To maximize power savings, the designer should apply these concepts when implementing any CMOS device.

### Derivation of Current Equation in CMOS Devices

Since the CMOS device is constructed of PMOS and NMOS transistors, the dynamic model is simply a capacitive structure for each transistor. Of course, the basic structure of a capacitor is the dielectric between two plates. In this case, the dielectric of the capacitor is the oxide layer on the silicon wafer and the plates are the poly or metal gate together with the inversion layer in the channel. The interconnecting metal/poly routing is also modeled as a capacitor where the plates are the routing itself together with any underlying routing or conductive material and the dielectric is any non-conductive structure between the plates. With these capacitive structures, the CMOS device is largely modeled as a collection of capacitors. Recall the basic equation for current through a capacitor:

$$I = C \cdot \frac{dV}{dt}$$

Breaking the derivative into its components we can extract the basic equation for frequency, which is the inverted value of a change in time, and simplify the equation:

$$I = C \cdot dV \cdot \frac{1}{dt}$$

$$I = C \cdot dV \cdot f$$

Since the voltage for capacitive structures in CMOS devices changes as a square wave with discontinuities between logic High and Low and is ideally rail to rail, we can further simplify the equation. For example, in a system of supply voltage, V, the change in voltage, dV, is V to 0V as shown and reduced here:

$$dV = V_2 - V_1$$

$$dV = V - 0$$

$$dV = V$$

Therefore the current equation for each capacitive structure becomes:

$$I = C \cdot V \cdot f$$

where:

- **I** = current in Amperes
- **C** = the capacitance of the capacitive structure in Farads
- **V** = the system voltage in Volts
- **f** = the toggle frequency of the capacitive structure in Hz

Dynamic device current is the summation of all capacitive structures toggling over time. Voltage remains the same for all equations and can be assumed to be a constant. A device of n capacitive structures can be represented as follows:

$$I_{Dynamic} = V \cdot \sum_{i=1}^{n} C_i \cdot f_i$$

To obtain total device current, static current must be added to the dynamic current:

$$I_{Total} = I_{Static} + V \cdot \sum_{i=1}^{n} C_i \cdot f_i$$

For illustrational purposes, it may be easier to discuss total current using a simplified version of this equation:

$$I_{CC} = I_{CCQ} + C \cdot V \cdot f$$

where:

- **I$_{CC}$** = *total* device current in Amperes
- **I$_{CCQ}$** = *quiescent* device current in Amperes
- **C** = the *lumped* capacitance of the device in Farads
- **V** = the system voltage in Volts
- **f** = the *average* device toggle frequency in Hz

Recall that to obtain power, multiply current by voltage to yield Watts.

## Reduce System Speed

It becomes obvious from the equation that for a fixed device capacitance and voltage, reducing the average device toggle rate will reduce power consumption.

Limiting the system clock speed as well as the data bus speed to slower values will reduce power consumption since average device toggle rate will become smaller. Careful analysis of

the required CPLD clock speed is essential for low power design. Over-clocking the CPLD design beyond the needs of the logic unnecessarily consumes extra power. Evaluating the minimum required speed of the CPLD logic will ensure that the system clock will have a minimal effect on power consumption.

## Drive Inputs to Standard Logic Levels

Power consumption will rise dramatically when a CMOS input buffer is not driven to known, standard logic levels, otherwise known as allowing the input to "float". A voltage level between standard logic levels causes the input buffer transistors, typically P and N channel, to be biased in a manner where both are in the ON state. When biased in this way, a large amount of current will flow between the power and ground supplies of the device via this channel. It is therefore imperative to drive the input buffer to a known logic level High or Low state to turn off one of these two transistors and avoid this situation.

The beauty of CMOS logic is that power consumption is nearly zero when logic gates are held at a known logic input level. FZP technology found in CoolRunner-II CPLDs uses this principle to provide dramatic power savings over other CPLDs.

Further, it is advised to drive the input to the full voltage rail on a High logic level and fully to ground on a logic Low level. Even though the voltage level is within the acceptable logic levels, i.e., $V_{IH}$ and $V_{IL}$, the closer the voltage is to the absolute voltage rails, the less current will be consumed in the input buffer. This effect is much less than the scenario where the input voltage floats between logic levels, but nevertheless can have a significant impact on total power consumption when summed across several I/Os.

## Increase Input Edge Speed

CMOS device inputs that are driven by a slowly switching source will consume more power since the input spends more time biased in the linear region. Again, the linear region is found when an input is biased at a voltage between standard logic High or Low levels. When the CMOS input buffer is biased in this manner, both the P and N channel transistors are turned on, allowing a relatively large amount of current to pass to ground. The longer the buffer is biased in this fashion, the more power will be consumed by the device. Therefore, it is recommended to quickly switch the signal as it is applied to the inputs of the CMOS device. This applies to all clock pins, dedicated input pins, or I/Os configured as inputs or bidirectional.

If implementing fast rise time edges is not an option for the specific application, it is advised to use Schmitt triggers to avoid the excessive power consumption inherent with slow rising signals.

## Eliminate Bus Conflicts

Occasionally, bus conflicts occur where two output buffers are driving a line at the same time in opposite directions. This adversely affects logic performance as well as power consumption. Two drivers attempting to swing the bus at opposite voltage levels will draw excessive current.

A similar situation occurs when a bus is pulled High via a pull-up resistor, for example, and the output buffer is driving the bus Low. In this example, current will flow from the power source, through the pull-up resistor, the N channel transistor in the output buffer, and to ground. Current, in this case, is a function of the value of the pull-up resistor and the N channel impedance. A weak pull-up resistor, on the order of 10k ohms or more, is a good place to start if it is desired to use such a component. The larger the resistor, the less power will be consumed, but this will slow down the bus response time when the resistor is required to charge the bus to the High level if the bus is in the 3-state condition.

It is recommended to continuously drive a bus line High or Low with one device at a time and remove pull-up resistors whenever possible to obtain the lowest power condition. In some situations, this may not be possible. For example, an SMBus or I2C SDA line is required to be released by all components but be at a High level when released. In this case, a pull-up resistor is required.

### Bus Terminations

A different case where pull down resistors are necessary is when shunt bus termination is required to reduce reflections in high speed designs. These terminations are usually designed to be pull down resistors whose value equals the equivalent impedance of the data bus transmission line and are positioned as close to the load pin as possible. Whenever the output buffer is driving a transmission line with shunt resistors, the P channel transistor will source current into the load and therefore will raise power consumption.

It may be possible to avoid using shunt termination resistors by using a very short transmission line. The rule of thumb for a short transmission line is one whose length is less than one-sixth the electrical length of the rise time, and is described using the following equation:

$$\text{Length} \ll \frac{1}{6} \cdot \frac{T_{rise}}{\sqrt{LC}}$$

where:

- **Length** = maximum line length in inches
- **T$_{rise}$** = rise time in seconds
- **L** = the line inductance in Henries/inch
- **C** = the line capacitance in Farads/inch

By using the CoolRunner-II CPLD slew rate feature configured to SLOW, the rise time becomes longer. Using the above equation, it can be determined if the length of the transmission line can be effectively long enough to avoid reflections altogether. If by using the SLOW slew rate feature, the length of the transmission line is short by the above rule of thumb, shunt bus termination resistors can be avoided thereby saving power.

If either method is not an option, insert a series termination resistor positioned at the source pin with the same value as the transmission line impedance. This option will avoid excessive power consumption (since there is no shunt load) and eliminate reflections at the source. However, a series termination resistor will allow one reflection at the load (since the load impedance does not match the transmission line impedance) which implies that any component mid way between the source and the load will see two transitions: the incident wave and the single reflected wave created at the load.

### Reduce System Voltage

Using the total current equation with fixed capacitance and average frequency, it becomes readily apparent that reducing system voltage will reduce power consumption, provided the system voltage remains within recommended operating specifications. Therefore, it makes sense to use a 1.8V device in lieu of a 3.3V or 2.5V device. CoolRunner-II CPLDs are 1.8V devices and therefore utilize this concept. Further, a device made with a smaller process technology, such as CoolRunner-II CPLDs, will generally have lower lumped internal capacitance values thereby reaching additional power savings. Combining these two factors with reducing average system frequency will also cut power consumption.

With any voltage device, there is a recommended operating range, and it may be advantageous to operate low in that voltage range to further reduce power consumption. Voltages that are outside the recommended operating range may cause excessive power consumption including adverse functional performance.

### Reduce Bus Loading

Connecting an external bus to a CMOS device will increase power consumption due to the loading effects of the bus on the device. The primary factor from loading is given by capacitive or resistive bus components as seen by the output buffer looking into the bus.

Capacitive loading comes in two forms: lumped and distributed. Lumped capacitance is typically found from the gate of the input buffer of other connected CMOS devices. It is also

developed from any capacitive element that is attached to the bus. Distributed capacitance is present due to the routing of the trace on the PCB. Both types will charge and discharge during logic transitions based on the previous logic state of the bus. Capacitive loading will draw current from the CMOS device whose output buffer is driving the bus, thereby increasing apparent power consumption of that device as seen at its power pins. To minimize power consumption based on this capacitive loading effect, it is necessary to reduce the size of the lumped capacitance found in attached devices, and to shorten the PCB traces. Doing so will also increase potential system speed since reflections are less likely.

Resistive loading is usually found when devices with a resistive element to their impedance is attached to the bus. For example, a pull down resistor attached to a PCB trace will allow for current to flow from the CMOS device power rail, through the P channel of the output buffer transistor, through the resistor, and to ground, increasing observed power consumption at the CMOS device power pins. Reducing resistive loads will reduce power consumption.

Keep in mind that many components are comprised of more than one type of impedance element. For example, capacitors also have resistive and inductive elements, albeit small.

### Further Power Saving Techniques

For further discussions regarding power saving techniques, particularly those involving logic design, review XAPP346 - Low Power Tips for CoolRunner Design found at http://www.xilinx.com/xapp/xapp346.pdf. Although the referenced application note discusses CoolRunner XPLA3 CPLDs, the basic principles apply to CoolRunner-II CPLDs.

## Conclusion

For the first time in the CPLD industry, CoolRunner-II products combine true high speed logic with ultra low power. Unique features of the CoolRunner-II CPLD, such as CoolCLOCK and DataGATE, allow the designer to further reduce power consumption. By using these features combined with good design practice, as outlined in this document, designers can be assured that their design will experience optimal low power benefits without sacrificing high speed.

## References

1. Johnson H. and Graham M. (1993): High-Speed Digital Design: A Handbook of Black Magic Prentice Hall PTR

## Revision History

The following table shows the revision history for this document.

| Date | Version | Revision |
|---|---|---|
| 05/08/02 | 1.0 | Initial Xilinx release. |