MAX II CPLD Design Guidelines

December 2007, Ver 1.1 Application Note 428

Introduction

With the flexibility of complex programmable logic devices (CPLDs), together with their low power consumption and low cost, more designers are using CPLDs in their system design. Using MAX® II CPLDs in your design can be very straightforward when you have some guidelines to follow, even if you are not a frequent CPLD user. This application note aims to provide the necessary guidelines on using MAX II devices in your design, and help you avoid some of the problems users frequently face.

The design guidelines in this application note are categorized under the following sections:

- "CPLD Selection" on page 1
- "Hardware Setup Checklist" on page 3
- "Design Checklist" on page 12
- "Additional Development Tools and References" on page 18

For detailed information about the MAX II device specification, refer to the *MAX II Device Handbook*.

CPLD Selection

The MAX II CPLD family offers various device densities to cater to different user needs. Here are some factors you should consider when choosing the MAX II device:

- Number of user I/Os and package offerings
- Logic density
- V_{CCINT} and power consumption
- Temperature grade

Number of User I/Os and Package Offerings

The MAX II family has a maximum of 272 input/output (I/O) pins. The devices are available in various packages and are suitable for different board requirements. The MAX II Micro FineLine BGA packages are designed for portable applications, where space-saving is important. MAX II devices support vertical migration within the same package, allowing an easy switch between different device densities. Table 1 shows the packages and the number of user I/O pins available for MAX II devices.

Table 1. MAX II Packages and User I/O Pins

Device	68-Pin Micro FineLine BGA *(1)*	100-Pin Micro FineLine BGA *(1)*	100-Pin FineLine BGA *(1)*	100-Pin TQFP	144-Pin TQFP	144-Pin Micro FineLine BGA *(1)*	256-Pin Micro FineLine BGA *(1)*	256-Pin FineLine BGA	324-Pin FineLine BGA
EPM240 EPM240G	—	80	80	80	—	—	—	—	—
EPM570 EPM570G	—	76	76	76	116	—	160	160	—
EPM1270 EPM1270G	—	—	—	—	116	—	212	212	—
EPM2210 EPM2210G	—	—	—	—	—	—	—	204	272
EPM240Z	54	80	—	—	—	—	—	—	—
EPM570Z	—	76	—	—	—	116	160	—	—

Note to Table 1:
(1) Packages available in lead-free versions only.

Logic Density

MAX II CPLDs have 240 to 2,210 logic elements (LEs), or typically 192 to 1,700 equivalent macrocells for you to implement different functions. For example, the EPM240 device has 240 LEs; this means it has 240 registers available in the device.

V_{CCINT} and Power Consumption

MAX II devices support 3.3 V or 2.5 V for V_{CCINT}. The devices have an internal linear voltage regulator, which regulates the external supply voltages to the internal operating voltage of 1.8 V. The MAX IIG and MAX IIZ devices do not use an internal voltage regulator, thus the devices operate at 1.8-V V_{CCINT}. Devices with lower V_{CCINT} have lower total power consumption.

For more information about the low power applications for MAX II devices, refer to *AN 422: Power Management in Portable Systems Using MAX II CPLDs*.

Temperature Grade

The MAX II family offers three different temperature grades: commercial, industrial, and extended temperature grades. Select the correct temperature grade according to your application. Table 2 shows the operating temperature range for devices of the three temperature grades.

Table 2. MAX II Device Operating Temperature Range *Note (1)*

Temperature Grade	Operating Temperature		Unit
	Minimum	Maximum	
Commercial	0	85	°C
Industrial	–40	100	°C
Automotive	–40	125	°C

Note to Table 2:
(1) MAX IIZ devices are available only in commercial grade.

For detailed information and specifications of the MAX II device family, refer to the *MAX II Device Handbook*.

Hardware Setup Checklist

This section lists some of the items you should check when considering your hardware setup.

- "VCCINT and VCCIO Voltages" on page 4
- "Power-Up Sequencing" on page 4
- "Input Pin Connection" on page 4
- "Unused Pin Connection" on page 5
- "Input Pin Voltages" on page 5
- "Output Pin Source Current" on page 6
- "JTAG Pins Pull Up/Down" on page 6
- "JTAG Chain Connection for Programming" on page 7
- "JTAG Chain Containing Devices with Different VCCIO" on page 7
- "JTAG Signal Buffering" on page 8
- "Device Output-Enable Pin" on page 9
- "Chip-Wide Reset" on page 10
- "Register Power-Up Level" on page 10
- "Latch-Up Prevention" on page 11

V_{CCINT} and V_{CCIO} Voltages

Ensure that the device is powered up within the recommended operating voltage range. Do not leave any V_{CCINT}, V_{CCIO}, or ground pins unconnected as this can cause current leakage. V_{CCIO} of all I/O banks and V_{CCINT} of the device must be fully powered up, not only for normal operation, but also for in-system programming (ISP).

The MAX II family has the MultiVolt™ core and I/O features. The MultiVolt core feature allows the device to support different V_{CCINT} voltages. The MAX II device accepts 2.5-V or 3.3-V V_{CCINT}, while MAX IIG and MAX IIZ devices accept 1.8-V V_{CCINT}. The MultiVolt I/O feature allows the device to support 1.5-V, 1.8-V, 2.5-V, and 3.3-V V_{CCIO} voltages. Each I/O bank is powered up individually by the VCCIO pins of that particular bank, and is independent of the V_{CCIO} of other I/O banks.

For information about the power regulation for MAX II devices, refer to *Power Management Reference Guide for Altera FPGAs & CPLDs* at www.altera.com/support/devices/vendors/pow-vendors.html.

Power-Up Sequencing

MAX II devices support hot-socketing. They are designed to operate in multiple-voltage environments, so they can tolerate any power-up sequence. You can either power up V_{CCINT} or V_{CCIO} first, or both at the same time. Input signals of 3.3, 2.5, 1.8, or 1.5 V can drive the devices without special precautions before V_{CCINT} or V_{CCIO} is applied. Normal operation does not occur until both power supplies are in their recommended operating range.

The *MAX II I/O Characteristics During Hot Socketing* white paper shows the I/O pin characteristics for different power-up sequences.

Input Pin Connection

All input pins of your design should be driven by either V_{CC} or ground. This is because floating input pins have undefined values and your design may not work correctly with those undefined input values. Floating input pins also cause additional noise going into the device. This applies to bidirectional pins functioning as input pins as well.

Unused Pin Connection

The Quartus® II software generates the pin report file (**.pin**) when you compile your design. This report file specifies how you should connect the unused pins of your device. For a MAX II device, unused I/O pins are marked in the report file as either:

- GND*
- RESERVED
- RESERVED_INPUT
- RESERVED_INPUT_WITH_WEAK_PULLUP
- RESERVED_INPUT_WITH_BUS_HOLD

depending on how you set the unused pins in the Quartus II software.

All I/O pins specified as GND* can either be connected to ground to improve the device's immunity to noise or left unconnected. Leave all RESERVED I/O pins unconnected on your board because these I/O pins drive out unspecified signals. Tying a RESERVED I/O pin to V_{CC}, ground, or another signal source can create contention that can damage the output driver of the device.

RESERVED_INPUT I/O pins can be connected to a high or low signal on the board while RESERVED_INPUT_WITH_WEAK_PULLUP and RESERVED_INPUT_WITH_BUS_HOLD pins can be left unconnected.

Input Pin Voltages

The voltage level of the input signal should meet the high-level (V_{IH}) and low-level (V_{IL}) input voltages of the device. The input pin may not recognize the input signal correctly if the voltage level of the signal falls between the minimum of V_{IH} and maximum of V_{IL}. Also, do not drive the pin outside the recommended input voltage (V_I) range (-0.5 V to 4 V).

The MultiVolt I/O feature allows the device to interface with systems of different supply voltages. Each I/O bank is powered up independently by the VCCIO pins of that bank. Assign the pins that work with the same voltage level in the same I/O bank so that you can use the other I/O banks for other V_{CCIO} voltages.

For more information about using a MAX II device in a multi-voltage system, refer to the *Using MAX II Devices in Multi-Voltage Systems* chapter of the *MAX II Device Handbook*.

Output Pin Source Current

Sourcing or sinking large amounts of current from output pins continuously, for example, by pulling a high output pin to ground or connecting a low output pin to V_{CC} directly, can damage the device due to electromigration, where the movement of the conducting metal's atoms is caused by the large amount of electric current flowing through it. The maximum current a MAX II I/O pin can sink is 25 mA. If certain pins must be pulled high or low, pull the pins through external resistors.

Using the device I/O pin to power up another device can source a large amount of current from the pin as well. Do not use I/O pins as a direct power source. If you need to use the MAX II device to control the power-up of another device, use the MAX II I/O pin to control a switch, for example, a relay or a transistor, that powers up the device.

In addition, do not source more than 170 mA of current for a set of I/O pins between any two V_{CCIO} pads, or sink more than 130 mA of current for a set of I/O pins between any two GNDIO pads. For example, the EPM240 device has six GNDIO pads, which provide six I/O regions that can sink up to 130 mA. If you need to sink 15 mA for the outputs, you can have eight outputs per region. With the six I/O regions between GNDIO pads, there are 48 possible outputs, each sinking 15 mA.

For information about the maximum sink and source current for the MAX II device, refer to *AN 286: Implementing LED Drivers in MAX & MAX II Devices*.

JTAG Pins Pull Up/Down

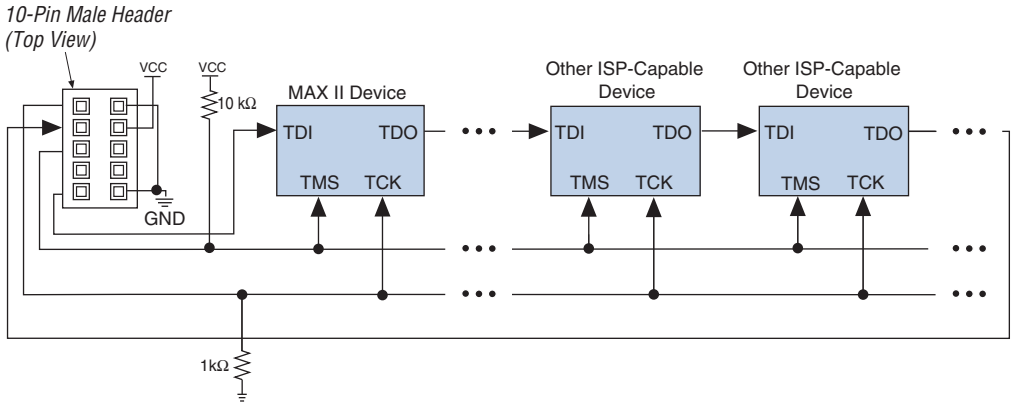
Noise at the JTAG pins, whether the device is in ISP or user mode, or during power-up, can cause the device to go into an undefined state or mode. Altera recommends pulling the `TCK` pin low and the `TMS` pin high through a 10-kΩ resistor.

The JTAG circuitry is activated when V_{CCINT} is powered up. If the `TMS` and `TCK` pins that are connected to V_{CCIO} and V_{CCIO} are not powered up, the JTAG signals are left floating. Any transition on the `TCK` pin can cause the JTAG state machine to transition to an unknown state, leading to incorrect operation when V_{CCIO} is finally powered up. To disable the JTAG state machine during power-up, the `TCK` pin should be pulled low to ensure that an inadvertent rising edge does not occur on `TCK` pin.

JTAG Chain Connection for Programming

Connect the JTAG pins of the device to the download cable header correctly. If you have more than one device in the chain, connect the TDO pin of a device to the TDI pin of the next device in the chain, as shown in Figure 1.

Figure 1. Devices in a JTAG Chain



To help you in debugging any JTAG programming issue, refer to the *JTAG Configuration & ISP Troubleshooter* in the Altera website at www.altera.com/cgi-bin/ts.pl?fn=jtagprog.

JTAG Chain Containing Devices with Different V_{CCIO}

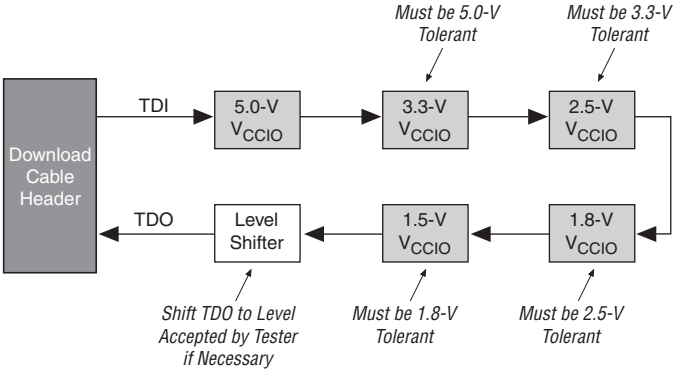
The operating voltage supplied to the Altera download cable by the target board through the 10-pin header determines the operating voltage level of the download cable. The JTAG pins for all MAX II devices reside in Bank 1 and their I/O standard support is controlled by the V_{CCIO} setting for Bank 1.

As the download cable interfaces with the JTAG pins of your device, ensure that the download cable operating voltage and the JTAG pin voltage are compatible. Refer to the datasheet for each download cable for the operating voltage.

In a JTAG chain containing devices with different V_{CCIO}, the devices with a higher V_{CCIO} level should drive the devices with the same or lower V_{CCIO} level. You only need one level shifter at the end of the chain with

this device arrangement. If this arrangement is not possible, you have to add more level shifters into the chain. Figure 2 shows the JTAG that contains devices with different V_{CCIO}.

Figure 2. Devices with Different V_{CCIO} in a JTAG Chain



JTAG Signal Buffering

The JTAG signal integrity determines the need to buffer a JTAG chain. Pay particular attention to the TCK signal because it is the JTAG clock and is the fastest switching signal compared to the other JTAG signals. Altera recommends buffering the signals at the connector because cables and board connectors tend to make bad transmission lines and introduce noise to the signals. After this initial buffer at the connector, add buffers as the chain gets longer or whenever the signals must cross a board connector.

At any time, when a cable must drive three or more devices, buffer the signal at the cable connector to prevent signal deterioration. Of course, this also depends on the board layout, loads, connectors, jumpers, and switches on the board. Anything added to the board that affects the inductance or capacitance of the JTAG signals increases the likelihood of a buffer to be added to the chain.

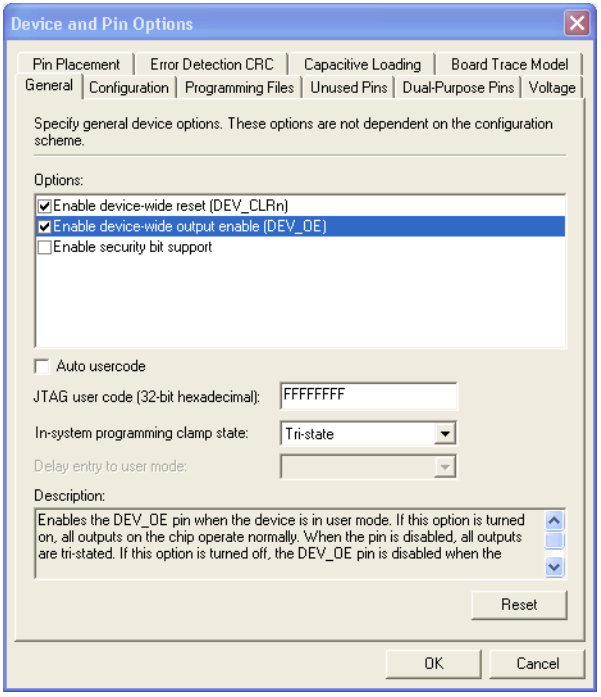
For the TCK and TMS signals that drive in parallel, each buffer should drive no greater than eight loads. If jumpers or switches are added to the path, decrease the number of loads.

Device Output-Enable Pin

The MAX II device has a chip-wide output-enable pin (DEV_OE) to control the output enable for all output pins in your design. This allows bus-sharing in your system where you can disable the MAX II output so that other devices can drive the bus. If you use this option, all outputs on the chip operate normally when DEV_OE is asserted. When the pin is deasserted, all outputs are tri-stated. The DEV_OE pin functions as a normal user I/O pin if the option is not used.

To set this in the Quartus II software, on the Assignments menu, click **Settings**. On the **Settings** dialog box, click **Device** and then click **Device and Pin Options**. Click the **General** tab and turn on **Enable device-wide output enable (DEV_OE)**. Set this option before compiling your design. Figure 3 shows the **Device and Pin Options** dialog box.

Figure 3. Device and Pin Options



Chip-Wide Reset

The MAX II device has a chip-wide reset pin (DEV_CLRn) for resetting all registers in the device. If this option in enabled, you can clear the device's registers by asserting this pin low because this chip-wide reset overrides all other control signals of the MAX II device.

In a system where not all the devices are powered up and start functioning simultaneously, you can hold this pin low before or during power-up to prevent the MAX II device from starting to function until the other devices go into user mode. The DEV_CLRn pin functions as a normal user I/O pin if this option is not used.

To set DEV_CLRn in the Quartus II software, on the Assignments menu, click **Settings**. On the **Settings** dialog box, click **Device** and then click **Device and Pin Options**. Click the **General** tab and turn on **Enable device-wide output enable (DEV_OE)**. Set this option before compiling your design. Figure 3 on page 9 shows the **Device and Pin Options** dialog box.

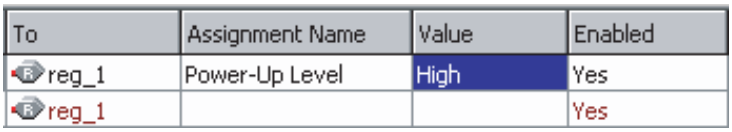
Register Power-Up Level

All registers are cleared upon power-up. By default, all registered outputs drive low when the device is powered up. In a system where you use the registered output to drive other devices, for example, controlling the reset of the devices, you might need the MAX II device output to drive high for a specific duration upon power-up until the whole system is properly initialized. After that, the registered output functions according to your design.

You can set the registered outputs to drive high upon power-up through the Quartus II software. The Quartus II software uses the "NOT Gate Push-Back" method to set the output high.

To set this in the Quartus II software, go to the Assignment Editor and set the **Power-Up Level** assignment for the register to **High**. Figure 4 shows the register power-up level setting in the Assignment Editor.

Figure 4. Register Power-Up Level

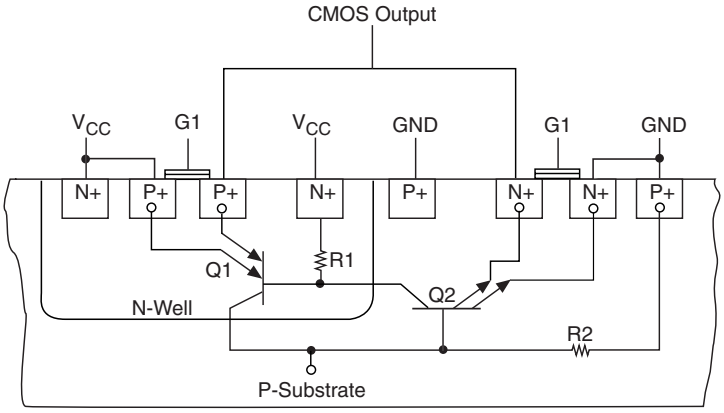


To	Assignment Name	Value	Enabled
reg_1	Power-Up Level	High	Yes
reg_1			Yes

Latch-Up Prevention

Parasitic bipolar transistors are present inside all CMOS devices. Under normal operating conditions, the base-emitter and base-collector junctions of these parasitic transistors, as shown in Figure 5, are not forward-biased and are not turned on. However, excessive current forced into or out of I/O pins, especially during power-up, can turn on those parasitic transistors.

Figure 5. Parasitic Bipolar Transistors in CMOS Devices



These parasitic transistors create destructive current paths in a device once they begin to conduct because the effect is regenerative and reinforces itself until potentially destructive currents are produced, thus causing the device to experience latch-up and draw heavy current until the device is powered down or damaged by high current flow.

MAX II devices support hot-socketing and can tolerate any power-up sequence for the V_{CCINT}, V_{CCIO}, and I/O pins. However, if your system permits, apply ground to the device first, then V_{CCINT} and V_{CCIO}, and finally the inputs. This can minimize the chances of inducing latch-up during power-up. For power-down, the power should be removed from the device in the reverse order—the inputs are removed first, then V_{CCINT} and V_{CCIO}, and finally ground.

For more information about latch-up, refer to the *Operating Requirements for Altera Devices* datasheet.

Design Checklist

This section lists some of the items you should pay attention to when you create your design.

- "Design Entry" on page 12
- "HDL Coding Style" on page 12
- "Global Clock" on page 13
- "Register Inputs" on page 13
- "Synchronous Design" on page 14
- "Schmitt Trigger Input" on page 15
- "Design Simulation" on page 15
- "Timing Violation" on page 16
- "Output Pin Current Strength and Slew Rate" on page 16
- "Pin Assignments" on page 16
- "Quartus II Design Assistant" on page 17

Design Entry

The Quartus II software allows you to create your design through schematic/block diagram or HDL coding. The commonly used HDL formats supported are Verilog and VHDL. For simple designs, using schematic/block diagram makes your task of creating the design easier. But for more complex designs, using HDL coding gives you the flexibility you need and makes your design more efficient.

The Quartus II software is able to generate the symbol or HDL files for the megafunctions you create and allows you to integrate the megafunction into your design, regardless of the design entry method.

The Quartus II software has an extensive component library that allows you to implement the functions of 74-series integrated circuits (IC), such as AND gates, OR gates, inverter, and flip-flops, and also the library of parameterized modules (LPM) where you can customize the modules according to your need.

HDL Coding Style

If you use HDL as the design entry for your design, pay attention to the coding style. HDL coding has a significant effect on the quality of results in terms of logic utilization and performance that you can achieve for your designs. Effective coding helps the synthesis tool to perform better when synthesizing your design.

Refer to the *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*.

Global Clock

Each MAX II device has four dual-purpose dedicated clock pins (GCLK[3..0]) that can be used to drive the global clock network for clocking or as normal I/O pins. The four global clock lines in the global clock network drive throughout the entire device. The global clock network provides clocks for all resources within the device including logic elements (LEs), LAB local interconnect, input/output elements (IOEs), and the user flash memory (UFM) block.

Assign the clock sources in your design to these pins so that you have a fixed and predictable delay for the clock signals. In the Quartus II Assignment Editor, assign the clock pins in your design to the device's clock pins. After that, under **Assignment Name**, choose **Global Signal**. Then under **Value**, select **Global Clock**. Enable this assignment. Figure 6 shows the global clock pin assignment in the Assignment Editor.

Figure 6. Global Clock Assignment

To	Assignment Name	Value	Enabled
clk_in	Location	PIN_12	Yes
clk_in	Global Signal	Global Clock	Yes
clk_in			Yes

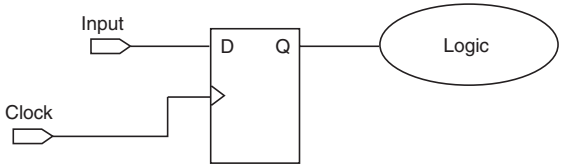
If you do not use these four clock pins to drive the global clock network, you can use them as general-purpose I/O pins.

Register Inputs

Input signals to your design may not always be stable and can have glitches or noise. To prevent the wrong signal from propagating into your design and affecting your system's functionality, have the input signal go through a register before going into the design, as shown in Figure 7. As the input of the register is only sampled and transferred to the design on every active edge of the clock, glitches, or instability of the input signal do not propagate further into the design.

Use either the MAX II device's internal oscillator or an external clock signal as the clock source of the register, as long as the clock signal has a higher frequency than the input signal. You can implement the registered input with the clock signal, regardless of whether your design is in coding or block diagram.

Figure 7. Register Design Input

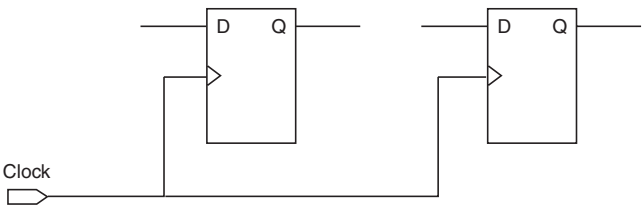


Synchronous Design

In a synchronous design, signal changes depend on the clock signal. On every active edge of the clock, the data inputs of registers are sampled and transferred to outputs.

Use a single clock source to clock the registers in your design, as shown in Figure 8. If two cascaded registers are triggered on different clock sources or edges, there is the risk that the second register will not have enough time to resolve the metastable output from the first register because of setup time violation, and thus clocking in an incorrect value.

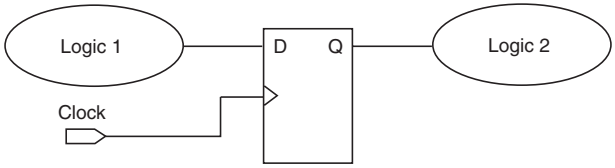
Figure 8. Synchronous Design



If the combinational logic output from your design feeds to another part of the design, have the signal go through a register, as in Figure 9. This applies if you are using the combinational logic output as a clock signal or as an asynchronous reset signal. Changes to the combinational output may trigger a period of instability due to propagation delays through the logic as the signal goes through a number of transitions before the output settles down to a new value.

As the input of the register is only sampled and transferred to the design on every active edge of the clock, changes happening on data inputs of the register do not affect the register output/input to the other part of the design until the next active clock edge. As long as the setup and hold time of the register are not violated, the register will effectively isolate any glitches or instable input signal from other logic.

Figure 9. Registering Combinational Output Signal



Also, your design should not rely on delay paths within the architecture of a device because any change in timing to a specific path could affect its functionality. Factors such as temperature, voltage, process change, or placement and routing change could affect the timing of logic paths in a device and cause unwanted functional changes. Synchronization eliminates unwanted functional changes.

For more information about synchronizing your design, refer to the *Design Recommendations for Altera Devices* chapter in volume 1 of the *Quartus II Handbook* or Quartus II Help.

Schmitt Trigger Input

For noisy input signals, use Schmitt trigger inputs. Schmitt triggers provide hysteresis on the input buffer, preventing slow-rising, noisy input signals from ringing or oscillating on the input signal driven into the logic array. This provides system noise tolerance on the device inputs, but adds a small, nominal input delay. The Schmitt trigger also allows input buffers to respond to slow input edge rates with a fast output edge rate.

Design Simulation

With the appropriate input vectors, the Quartus II software allows you to simulate your design with either functional or timing simulation. Functional simulation allows you to check whether your design is logically functioning. This is suitable for designs that are not time- critical, for example, combinational designs or designs that use low frequency clocks.

Timing simulation takes the device timing information into consideration. The simulation result reflects the device operation more accurately. If your design uses a high-speed clock, it is important to run the timing simulation to ensure that your design works.

Timing Violation

The Quartus II Timing Analyzer checks for any timing violation during compilation. The timing analysis report shows the f_{MAX}, t_{SU}, t_{CO}, t_H, and t_{PD} for your design. Check for any warning or timing violation.

For more information about the timing analysis, refer to the *Quartus Timing Analysis* section in volume 3 of the *Quartus II Handbook*.

Output Pin Current Strength and Slew Rate

The current or drive strength determines the current of the output pin, while the slew rate determines the rise and fall time of an output signal.

To change the output pin drive strength, go to the Quartus II Assignment Editor and select **Current Strength**. Choose either **Maximum Current** or **Minimum Current**. You can also turn on or off the **Slow Slew Rate** feature in the Assignment Editor. By default, it is turned off. Figure 10 shows an example of Current Strength and Slow Slew Rate settings in the Assignment Editor.

Figure 10. Current Strength and Slow Slew Rate Settings

To	Assignment Name	Value	Enabled
out_pin	Current Strength	Minimum Current	Yes
out_pin	Slow Slew Rate	On	Yes
out_pin			Yes

For the actual current strength for the maximum and minimum settings, refer to the *MAX II Device Handbook*.

Pin Assignments

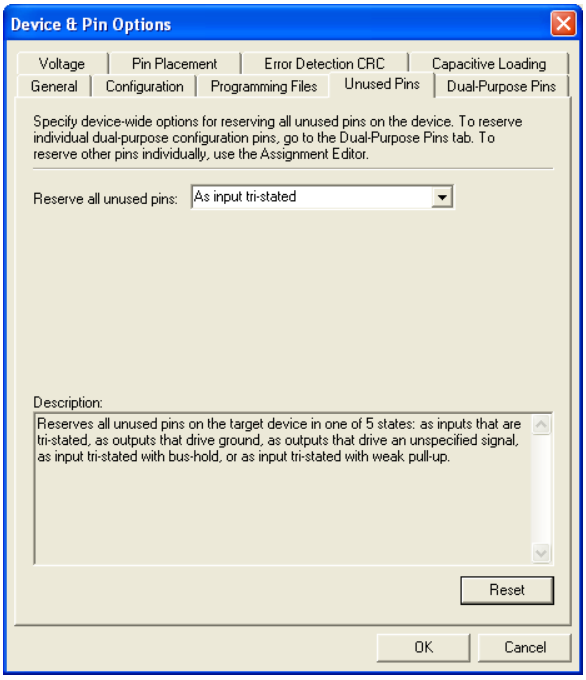
Sometimes the design does not work correctly because you have not made the correct pin assignments in the Quartus II software—especially for input pins! It is extremely critical that you assign all the used pins to the correct locations. If left unassigned, the Quartus II software automatically assigns the pins to certain locations. Your design may not function and worse still, this may cause contention.

For unused pins, you can set them to be either output pins driving unspecified signals, output pin driving ground, or tri-stated input pins (with or without weak pull-up or bus-hold) in the Quartus II software. If you connect these pins to other devices on the same board, it is best to set

the pins as tri-stated input pins to prevent them from affecting other devices. Tri-stated inputs do not drive out and only sink or source a maximum current of 10 µA. Setting the unused pins to the wrong state may cause contention and can damage the device.

To set the pins in the Quartus II software, on the Assignments menu, click **Settings**. On the **Settings** dialog box, click **Device** and then click **Device and Pin Options**. Click the **Unused Pins** tab and reserve all unused pins as input tri-stated, as shown in Figure 11.

Figure 11. Reserve All Unused Pins as Input Tri-Stated

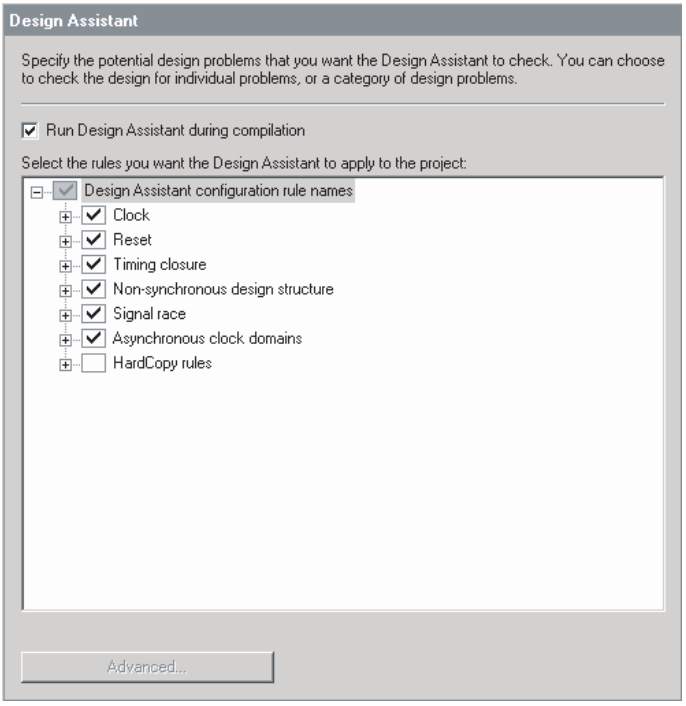


Quartus II Design Assistant

The Quartus II Design Assistant checks the reliability of a design based on a set of design rules during the design compilation. Some of the areas the Design Assistant checks are clocks, resets, timing closure, non-synchronous design structure, and so on. You can select the areas you want the Design Assistant to check.

To turn on the Design Assistant, on the Assignments menu, click **Settings** and then click **Design Assistant**. Figure 12 shows the areas the Design Assistant checks.

Figure 12. Quartus II Design Assistant



Additional Development Tools and References

Some additional development tools and references to help you use the MAX II devices are:

- MAX II Development Kit
- MAX II PowerPlay Early Power Estimator
- MAX II Device Family Errata Sheet

MAX II Development Kit

The MAX II Development Kit includes:

- MAX II Development Board with an EPM1270F256 device
- Quartus II Web Edition Development Software
- MegaCore® IP functions for the pci_t32 MegaCore

- Reference and demo designs

This development kit provides you with everything you need to prototype and develop complete solutions using MAX II devices. The reference and demo designs demonstrate the MAX II device's capability and provide a starting point to create your own custom designs.

For more information on the MAX II development kit, refer to the *MAX II Development Kit Getting Started User Guide* and the *MAX II Development Board Data Sheet*.

MAX II PowerPlay Early Power Estimator

The power consumption of your MAX II device depends on the resource usage of your design. The PowerPlay Early Power Estimator spreadsheet allows you to estimate the device power consumption and provides the thermal analysis data of the device.

The PowerPlay Early Power Estimator allows you to enter information into sections based on MAX II architectural features, for example, clocks, logic core, UFM, and I/Os, and provides the estimated power consumption for each of the areas.

You can download the MAX II PowerPlay Early Power Estimator spreadsheet from www.altera.com. For more information on the MAX II PowerPlay Early Power Estimator, refer to the *Understanding and Evaluating Power in MAX II Devices* chapter of the *MAX II Device Handbook*.

MAX II Device Family Errata Sheet

The *MAX II Device Family Errata Sheet* explains the power condition requirement for older MAX II devices. Also, the errata sheet explains some limitations of the EPM1270 engineering sample (ES) devices. All these issues are fixed in the later revisions of the MAX II devices.

Conclusion

With the necessary information and guidelines, using CPLDs in your system can be an easy task. This application note is written based on the problems users commonly face when using CPLDs and provides a checklist and easy-to-follow guidelines to ensure the correct operation of Altera MAX II devices.

Referenced Documents

This application note references the following documents:

- *AN 286: Implementing LED Drivers in MAX & MAX II Devices*
- *AN 422: Power Management in Portable Systems Using MAX II CPLDs*
- *Design Recommendations for Altera Devices* chapter in volume 1 of the *Quartus II Handbook*
- *MAX II Development Kit Getting Started User Guide*
- *MAX II Development Board Data Sheet*
- *MAX II Device Family Errata Sheet*
- *MAX II Device Handbook*
- *MAX II I/O Characteristics During Hot Socketing* white paper
- *Operating Requirements for Altera Devices Data Sheet*
- *Quartus Timing Analysis* section in volume 3 of the *Quartus II Handbook*
- *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*
- *Understanding and Evaluating Power in MAX II Devices* chapter of the *MAX II Device Handbook*
- *Using MAX II Devices in Multi-Voltage Systems* chapter of the *MAX II Device Handbook*

Document Revision History

Table 3 shows the revision history for this document.

Table 3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
December 2007 v1.1	• Updated Table 1. • Added *Note (1)* to Table 2. • Removed Table 3 of previous version. • Updated "VCCINT and VCCIO Voltages" section with MAX IIZ information. • Added "Hardware Setup Checklist" and "Design Checklist". • Added "Referenced Documents" section.	—
September 2006 v1.0	Initial release	—

Hardware Setup Checklist

This checklist provides a summary of the guidelines described in this document. Use the checklist to verify that you have followed the guidelines for each stage of your design.

Project Name:
Date:

"VCCINT and VCCIO Voltages" on page 4

	Done	N/A	
1	☐	☐	Power up the device within recommended operating voltage range.
2	☐	☐	Do not leave the V_{CCINT}, V_{CCIO}, or ground pin unconnected.

"Input Pin Connection" on page 4

	Done	N/A	
3	☐	☐	Ensure that all input pins including bidirectional input pins are driven by either V_{CC} or ground. Floating input pins have undefined values and may cause additional noise.

"Unused Pin Connection" on page 5

	Done	N/A	
4	☐	☐	Connect all GND pins to ground to improve the device's immunity to noise.
5	☐	☐	Leave all RESERVE I/O pins unconnected as these I/O pins drive out unspecified signals.

"Input Pin Voltages" on page 5

	Done	N/A	
6	☐	☐	Voltage level should meet the high-level (V_{IH}) and low-level (V_{IL}) input voltages of the device. Do not drive pin outside the recommended input voltage (V_I) range.
7	☐	☐	Assign pin that works with the same voltage level in the same I/O bank so that you can use the other I/O banks for other V_{CCIO} voltages.

"Output Pin Source Current" on page 6

	Done	N/A	
8	☐	☐	Do not tie output or input pin directly to GND or V_{CC}. Sourcing or sinking large amounts of current from output pins continuously can damage the device.
9	☐	☐	If certain pins need to be pulled high or low, pull the pins through external resistors.

"JTAG Pins Pull Up/Down" on page 6

	Done	N/A	
10	☐	☐	Pull the TCK pin low and the TMS pin high through a 10-kΩresistor to disable the JTAG state machine during power-up.

"JTAG Chain Connection for Programming" on page 7

	Done	N/A	
11	☐	☐	Connect the JTAG pins of the device to the download cable header correctly.
12	☐	☐	Connect the TDO pin of a device to the TDI pin of the next device if you have more than one device in the chain.

"JTAG Chain Containing Devices with Different VCCIO" on page 7

	Done	N/A	
13	☐	☐	Ensure that the download cable operating voltage and the JTAG pin voltage are compatible.
14	☐	☐	Ensure that the device with a higher V_{CCIO} level should drive the device with the same or lower V_{CCIO} level in a JTAG chain that contains devices of different V_{CCIO}.

"JTAG Signal Buffering" on page 8

	Done	N/A	
15	☐	☐	It is recommended to buffer the signals at the connector because cables and board connectors tend to make bad transmission lines and introduce noise to the signal.
16	☐	☐	Add buffers other than the initial buffer when the device chain gets longer or whenever the signals must cross a board connector.

"Device Output-Enable Pin" on page 9

	Done	N/A	
17	☐	☐	Enable chip-wide output-enable DEV_OE pin to control output enable for all output pins in your design. All outputs operate normally when DEV_OE is asserted. When the pin is deasserted, all outputs are tri-stated.

"Chip-Wide Reset" on page 10

	Done	N/A	
18	☐	☐	Enable chip-wide reset DEV_CLRn pin to reset all the registers in the device. This pin overrides all other control signals of the MAX II device.

"Register Power-Up Level" on page 10

	Done	N/A	
19	☐	☐	Depending on the design, set the output to drive high in the Quartus II software for a specific duration upon power-up or until the system is properly initialized.

"Latch-Up Prevention" on page 11

	Done	N/A	
20	☐	☐	If your system permits, apply ground to the device first, then V_{CCINT} and V_{CCIO}, and finally inputs to minimize the chances of inducing latch-up during power-up.
21	☐	☐	Apply the reverse order for power-down: the inputs are removed first, then V_{CCINT} and V_{CCIO}, and finally ground.

Design Checklist

This checklist provides a summary of the guidelines described in this document. Use the checklist to verify that you have followed the guidelines for each stage of your design.

Project Name:
Date:

"Design Entry" on page 12

	Done	N/A	
1	☐	☐	Consider the trade-off between schematics or HDL for your design entry based on your design's complexity.
2	☐	☐	Other than the Quartus II software, consider building your design with third-party EDA tools, SOPC Builder, DSP Builder, or IP cores.

"HDL Coding Style" on page 12

	Done	N/A	
3	☐	☐	If you are using HDL for design entry, use the recommended coding styles.
4	☐	☐	If you are using third-party synthesis tools to synthesize your design, use the netlist generated by the synthesis tools for the Quartus II Fitter to do the place-and-route procedure.

"Global Clock" on page 13

	Done	N/A	
5	☐	☐	Pay attention to your design's clocks. Use clock pins, the global clock network, clock control blocks, and the PLL for your clock signals.
6	☐	☐	Assign the clock sources in your design to clock pins so that you will have a fixed and predictable delay for the clock signals.

"Register Inputs" on page 13

	Done	N/A	
7	☐	☐	To prevent the wrong signal from propagating into your design and affecting your system's functionality, have the input signal go through a register before going into the design.
8	☐	☐	Use either the MAX II device's internal oscillator or an external clock signal as the clock source of the register.

"Synchronous Design" on page 14

	Done	N/A	
9	☐	☐	Use a single clock source to clock the registers in your design.

10	☐	☐	For combinational logic that feeds to another part of the design, for example, as a clock signal or as an asynchronous reset signal, route the signal through a register.
11	☐	☐	Ensure that the setup and hold time of the register are not violated in order for the register to be able to isolate any glitches or instable input signal from other logic.

"Schmitt Trigger Input" on page 15

	Done	N/A	
12	☐	☐	Use Schmitt trigger inputs for noisy input signals.

"Design Simulation" on page 15

	Done	N/A	
13	☐	☐	Assign the appropriate input vector to allow the Quartus II software to simulate your design with either functional or timing simulation.

"Timing Violation" on page 16

	Done	N/A	
14	☐	☐	Use the Quartus II Timing Analyzer to check for timing violation.

"Output Pin Current Strength and Slew Rate" on page 16

	Done	N/A	
15	☐	☐	Use the correct current strength and slew rate on output or bidirectional pins to prevent signal overshoot or undershoot for signal noise reduction, or to prevent stair-step output from the pins.

"Pin Assignments" on page 16

	Done	N/A	
16	☐	☐	Ensure all pin assignments are assigned correctly especially the input pins.
17	☐	☐	Set the unused pin to either output pin driving unspecified signals, output pin driving ground, or tri-stated input pins (with or without weak pull-up or bus hold) in the Quartus II software.

"Quartus II Design Assistant" on page 17

	Done	N/A	
18	☐	☐	Use the Quartus II Design Assistant to check design reliability.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support/
Literature Services:
literature@altera.com