Avoiding PCB Design Mistakes in FPGA-Based Systems

System design using FPGAs is significantly different from the regular ASIC and processor-based system design. In this white paper, we will examine some of the contributing factors, and more importantly, provide you with the key criteria to design FPGA-based systems successfully.

FPGAs have come a long way since they were introduced in the 1980s. Initially, FPGAs were used to create "glue logic," an essential function not performed by other components on the board. Today, FPGAs have the density and functionality to implement an entire system on chip, as shown in Figure 1.

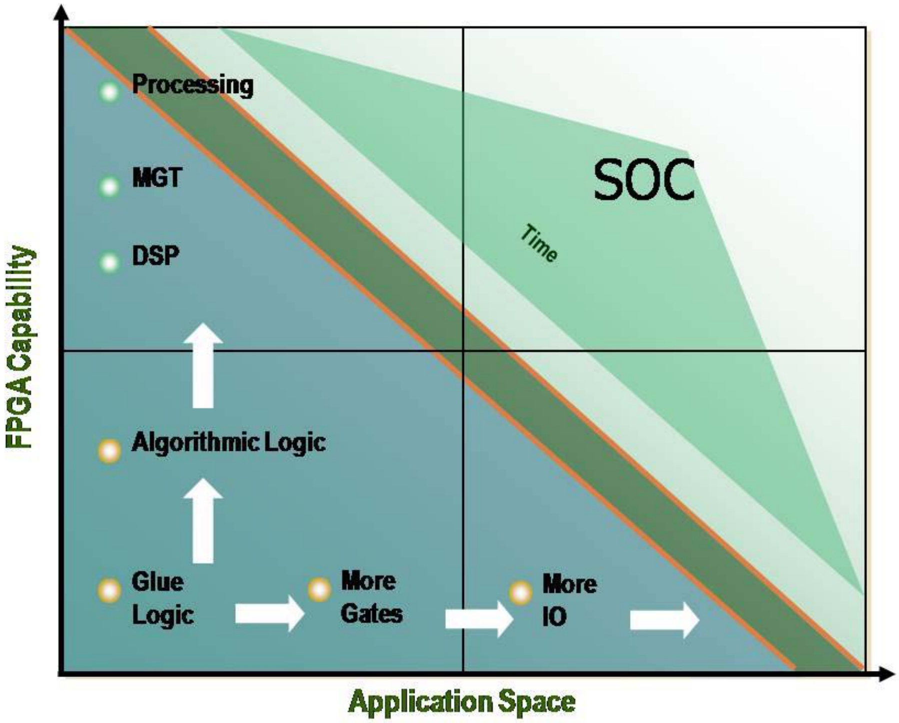


Figure 1 - FPGA Growth and Usage Trends

This increase in FPGA capability, combined with the fact that systems have become more complex, has introduced several FPGA board design challenges. The reason for this added complexity in FPGA-based boards is that the current board tools have not kept pace with the growth in FPGAs. These generic tools are used for designing PCBs containing components with non-programmable pins such as processors and ASICs.

Designing High-Quality FPGA-Based Systems

When your goal is first-time success, high quality, and minimal debug effort, it means a laundry list of items that should be considered—a list that is especially long for FPGA-based systems! Today, this means a lot of busy work and attention to minute details.

System-level design problems can be classified into the following categories:

- Functional – Physical issues that cause the design to not work. For example, if a clock to the FPGA is not coming in through a *clock pin*, the clock cannot be distributed with a reasonable skew, and hence the design will fail to function.
- Electrical – Electrical issues that cause the board to not work. For example, if a 3.3V LVTTL signal is connected to an FPGA bank with a 1.8V voltage rail, the signal will not electrically function, and so the design will fail.
- Marginal – Issues that allow the board to work most of the time, but not all of the time. For example, if a clock is connected to a non-clock pin in the FPGA, it cannot be distributed correctly within the FPGA. When this happens, the design may not work at some frequencies, though it may work at certain other frequencies.

Some of the electrical and functional issues require a re-spin of the final PCB. These may be rectified with "green wires" (or wires of your favorite color!), but may make the PCB less robust. Such fixes typically require a re-spin before the board is production worthy. Marginal issues are not only harder to find, but are harder to debug and fix as well. More importantly, the marginal issues are the ones that could show up at a customer site—which means these issues should be avoided at all costs!

Power Supplies and Power Distribution Systems

FPGAs have a high number of power supplies. For example, a simple design with a PCI Express (PCIe) interface and a DDR2 DIMM requires the power supplies shown in Table 1.

Voltage rail value	Need
1.2V	Serial I/O power
0.9V	Reference voltage for DDR2
3.3V	LVTTL control signals for PCIe interface
1.8V	I/O voltage for FPGA banks connected to DDR2
1.0V	FPGA internal voltage
GND	Ground

Table 1 - Power Supplies for a Simple Design

Every bank of an FPGA can require two or more power supplies. Moreover, the logical connections to the bank determine the voltage and current requirements for the power rail voltage and the current requirement for that bank. This means that you should pay attention to your voltage connections every time there is even a seemingly “minor“ change in the pins connected to the bank.

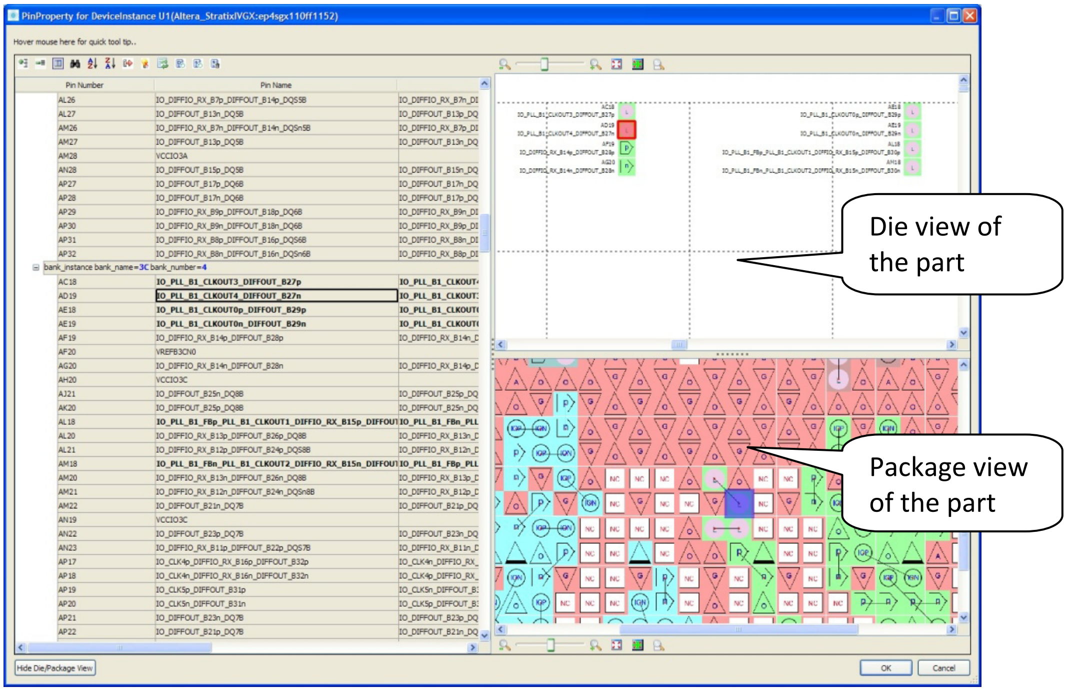
For example, if a DDR2 SDRAM is connected to a particular bank, a 1.8V and 0.9V (reference voltage) power supply is required for that bank. On the other hand, if an LVDS bus is connected to the bank, a 2.5V power supply is required to be connected instead of 1.8V. If you consider a DDR2 DIMM, most pins in the DIMM use 1.8V SSTL. However, some I^2C signals can be any standard. Various decoupling methods are recommended for each of these different power supplies.

To summarize, the power-supply connections require the following considerations:

- Create a power budget for each power supply voltage in your system. This ensures that the power supplies can adequately supply the current required by the FPGA and the interface.
- Review to ensure the power rails are of the right value. This may seem an obvious statement, but it is important because last-minute logic changes and pin optimizations may change the power rail requirements.
- Add decoupling caps recommended by Altera.
- Ensure that power-supply sensors are located close to the FPGA power pins to ensure the power supply can drive the dynamic current required by the FPGA.
- Follow the recommendations of Altera and the power supply vendors.

Clocks

FPGAs have different types of clock pins. For example, "global" clock pins distribute the clock throughout the FPGA, while other clock pins distribute clocks within a certain region of the FPGA. In Figure 2, global clocks are highlighted in red in the die view, in blue in the package view, and are bolded in the pin listing.


PinProperty for DeviceInstance U1(Altera_StratixIVGX:ep4sgx110ff1152)
Hover mouse here for quick tool tip..
Die view of the part
Package view of the part
Hide Die/Package View
OK
Cancel

Pin Number	Pin Name	
AL26	IO_DIFFIO_RX_B7p_DIFFOUT_B14p_DQS5B	IO_DIFFIO_RX_B7n_DI
AL27	IO_DIFFOUT_B13n_DQ5B	IO_DIFFOUT_B13p_DQ
AM26	IO_DIFFIO_RX_B7n_DIFFOUT_B14n_DQSn5B	IO_DIFFIO_RX_B7p_DI
AM27	IO_DIFFOUT_B13p_DQ5B	IO_DIFFOUT_B13n_DQ
AM28	VCCIO3A	
AN28	IO_DIFFOUT_B15p_DQ5B	IO_DIFFOUT_B15n_DQ
AP27	IO_DIFFOUT_B17p_DQ6B	IO_DIFFOUT_B17n_DQ
AP28	IO_DIFFOUT_B17n_DQ6B	IO_DIFFOUT_B17p_DQ
AP29	IO_DIFFIO_RX_B9p_DIFFOUT_B18p_DQ6B	IO_DIFFIO_RX_B9n_DI
AP30	IO_DIFFIO_RX_B9n_DIFFOUT_B18n_DQ6B	IO_DIFFIO_RX_B9p_DI
AP31	IO_DIFFIO_RX_B8p_DIFFOUT_B16p_DQS6B	IO_DIFFIO_RX_B8n_DI
AP32	IO_DIFFIO_RX_B8n_DIFFOUT_B16n_DQSn6B	IO_DIFFIO_RX_B8p_DI
⊟ bank_instance bank_name=3C bank_number=4		
AC18	**IO_PLL_B1_CLKOUT3_DIFFOUT_B27p**	**IO_PLL_B1_CLKOUT**
AD19	**IO_PLL_B1_CLKOUT4_DIFFOUT_B27n**	**IO_PLL_B1_CLKOUT**
AE18	**IO_PLL_B1_CLKOUT0p_DIFFOUT_B29p**	**IO_PLL_B1_CLKOUT**
AE19	**IO_PLL_B1_CLKOUT0n_DIFFOUT_B29n**	**IO_PLL_B1_CLKOUT**
AF19	IO_DIFFIO_RX_B14p_DIFFOUT_B28p	IO_DIFFIO_RX_B14n_D
AF20	VREFB3CN0	
AG20	IO_DIFFIO_RX_B14n_DIFFOUT_B28n	IO_DIFFIO_RX_B14p_D
AH20	VCCIO3C	
AJ21	IO_DIFFOUT_B25n_DQ8B	IO_DIFFOUT_B25p_DQ
AK20	IO_DIFFOUT_B25p_DQ8B	IO_DIFFOUT_B25n_DQ
AL18	**IO_PLL_B1_FBp_PLL_B1_CLKOUT1_DIFFIO_RX_B15p_DIFFOUT**	**IO_PLL_B1_FBn_PLL**
AL20	IO_DIFFIO_RX_B13p_DIFFOUT_B26p_DQ8B	IO_DIFFIO_RX_B13n_D
AL21	IO_DIFFIO_RX_B12p_DIFFOUT_B24p_DQS8B	IO_DIFFIO_RX_B12n_D
AM18	**IO_PLL_B1_FBn_PLL_B1_CLKOUT2_DIFFIO_RX_B15n_DIFFOUT**	**IO_PLL_B1_FBp_PLL**
AM20	IO_DIFFIO_RX_B13n_DIFFOUT_B26n_DQ8B	IO_DIFFIO_RX_B13p_D
AM21	IO_DIFFIO_RX_B12n_DIFFOUT_B24n_DQSn8B	IO_DIFFIO_RX_B12p_D
AM22	IO_DIFFOUT_B21n_DQ7B	IO_DIFFOUT_B21p_DQ
AN19	VCCIO3C	
AN22	IO_DIFFOUT_B23p_DQ7B	IO_DIFFOUT_B23n_DQ
AN23	IO_DIFFIO_RX_B11p_DIFFOUT_B22p_DQS7B	IO_DIFFIO_RX_B11n_D
AP17	IO_CLK4p_DIFFIO_RX_B16p_DIFFOUT_B32p	IO_CLK4n_DIFFIO_RX_
AP18	IO_CLK4n_DIFFIO_RX_B16n_DIFFOUT_B32n	IO_CLK4p_DIFFIO_RX_
AP19	IO_CLK5p_DIFFOUT_B31p	IO_CLK5n_DIFFOUT_B
AP20	IO_CLK5n_DIFFOUT_B31n	IO_CLK5p_DIFFOUT_B
AP21	IO_DIFFOUT_B23n_DQ7B	IO_DIFFOUT_B23p_DQ
AP22	IO_DIFFOUT_B21p_DQ7B	IO_DIFFOUT_B21n_DQ

Figure 2 - Illustration of Global and Regional Clock Pins in a Stratix IV FPGA

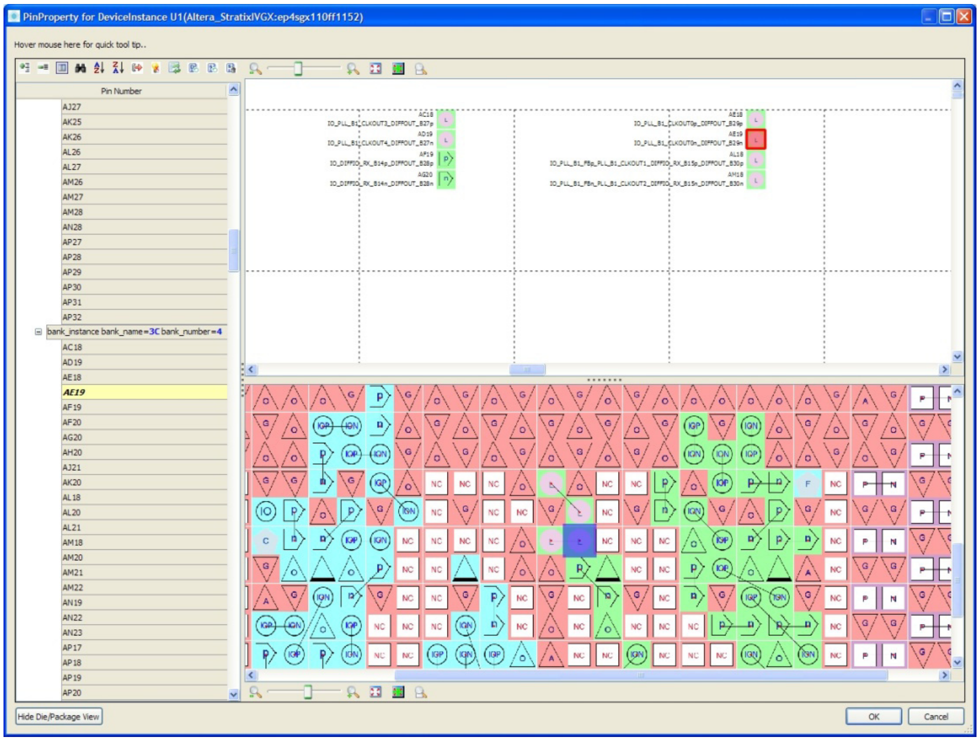
Selecting an incorrect pin for a clock can create a marginal design, which will work for some conditions, but not for others.

Selecting clock pins for your design requires the following considerations:

- Frequency of the clock – The frequency determines if you should use a single-ended clock or a differential clock.
- Local versus Global – If your clock is going to be used throughout the FPGA, you will need to use a global clock. If you are using the clock for local data capture, you may want to use one of the regional or local clock pins.

Source-Synchronous Buses

Source-synchronous buses are related to clocks, in that they contain clocks and an associated set of data signals. Most FPGAs require certain groups of pins to be used for source-synchronous buses, so only by selecting the correct FPGA pins can you ensure the clocks will reach the associated data signals. Figure 3 shows an illustration of different clock regions in an Altera Stratix IV FPGA with different colors in the die and package. In this view, the clock (phase-locked loop (PLL)) pins are represented with the letter "L" in a gray circle inside a green square.



PinProperty for DeviceInstance U1(Altera_StratixIVGX:ep4sgx110ff1152)

Hover mouse here for quick tool tip..

Pin Number
AJ27
AK25
AK26
AL26
AL27
AM26
AM27
AM28
AN28
AP27
AP28
AP29
AP30
AP31
AP32
⊟ bank_instance bank_name=3C bank_number=4
AC18
AD19
AE18
AE19
AF19
AF20
AG20
AH20
AJ21
AK20
AL18
AL20
AL21
AM18
AM20
AM21
AM22
AN19
AN22
AN23
AP17
AP18
AP19
AP20

AC18
IO_PLL_B1_CLKOUT3_DIFFOUT_B27p
AD19
IO_PLL_B1_CLKOUT4_DIFFOUT_B27n
AF19
IO_DIFFIO_RX_B14p_DIFFOUT_B28p
AG20
IO_DIFFIO_RX_B14n_DIFFOUT_B28n
AE18
IO_PLL_B1_CLKOUT0p_DIFFOUT_B29p
AE19
IO_PLL_B1_CLKOUT0n_DIFFOUT_B29n
AL18
IO_PLL_B1_FBp_PLL_B1_CLKOUT1_DIFFIO_RX_B15p_DIFFOUT_B30p
AM18
IO_PLL_B1_FBn_PLL_B1_CLKOUT2_DIFFIO_RX_B15n_DIFFOUT_B30n
Hide Die/Package View
OK
Cancel

Figure 3 - Illustrating Different Clock Regions With Different Colors

Your simulations and verification will not be able to determine a problem with your pin selections. For example, an unreported problem is a data pin outside the source-synchronous group so the associated clock cannot capture it. The FPGA tools report problems associated with clock and data groups when you place and route the corresponding register transfer level (RTL).

High-Speed Serial Buses

Today's FPGAs provide pins with high-speed transceivers for you to design interfaces with speeds up to 10 Gbps. Designing with high-speed serial I/Os requires you to follow the guidelines related to board layout, power supply designs, terminations, etc., very carefully. Some FPGAs require isolating the transceiver power-supply pins from any of the regular or single-ended I/Os. Not following these guidelines may cause marginal designs or a design that fails in the lab.

Achieving High Quality

The programmable nature of FPGAs makes them very different compared to ASICs and ICs. FPGAs have a very high number of I/O pins, certainly higher than the typical IC on the board. Higher numbers of I/Os make it more complex to design the PC board, and require special considerations to achieve high quality.

Let us try to define “high quality”:

- The board works consistently with a 20% margin in frequency, required both below and above the operating frequency.
- The board works when tested with key parts from different process corners.
- The board works with a 5% or 10% margin on voltage and temperature.

Here are some ways to achieve high quality:

- Reduce the number of PCB layers to improve the electrical quality of the signals.
- Ensure good return current paths for all the signal transmission paths, achieved by ensuring an adjacent power/ground plane for all the signal planes.
- Reduce the number of vias in the design to reduce signal inductance, thus transmitting and receiving the signal at a higher quality.
- Keep the trace length of signals to a minimum to reduce transmission line losses, thus improving signal quality.
- Budget your power supplies and ensure they are sufficient to handle your system needs.
- Ensure sufficient decoupling capacitors of the appropriate values are placed on the power supply.
- Terminate your high-speed signals based on Altera’s recommendations.
- Perform signal integrity (SI) simulations on your critical signals.
- Follow Altera’s guidelines for high-speed serial I/O designs.

Most importantly:

- Ensure you have a highly experienced board designer familiar with the unique nature of FPGAs and review your design thoroughly at every step.
- Review the final layout and the schematics to be sure they are consistent with the FPGA design and pin-outs.
- Ensure good communication between the PCB design team and FPGA design team.

7Circuits

7Circuits, an EDA tool from Taray, addresses the issues listed in this white paper. 7Circuits has been used very successfully by large OEMs to design FPGA-based boards. For more information, please visit www.tarayinc.com and request a free download of 7Circuits. You also can view a short demonstration of 7Circuits.

Taray, Inc., founded in 2002, provides industry-leading solutions for complex FPGAs used in system design. Their flagship product, 7Circuits, utilizes unique and patented I/O synthesis technology that optimizes and assigns the exploding programmable I/O complexity associated with FPGAs in system design. Taray's I/O synthesis automation enables faster time-to-market and better quality of system results for FPGAs used in system design.