

Summary

The advanced architecture of the XC9500 family, combined with consistent packaging options makes it easy to move an XC9500 design into a larger or smaller device and still keep the original footprint. This application brief describes how to prepare designs for easy migration and provides examples demonstrating how to verify their functionality.

Xilinx Family

XC9500

Introduction

CPLD design requirements often change, even after PC boards are laid out. Therefore, it is often advantageous to move the modified design into a different device within the XC9500 family, a device that more efficiently meets the needs for macrocell resources. With a little planning, it is easy to migrate an XC9500 design within the family and still keep the original pinouts. Thus, expensive and time consuming PC board re-work can be avoided, decreasing time to market. In addition, moving to a smaller device when design requirements are reduced, reduces the overall component cost of the system.

The advanced design migration capability of the XC9500 family gives unprecedented flexibility for:

- **Maintaining Designs in the Field** - Significant changes can be made without redesigning PC boards.
- **Reducing Cost** - If the redesign uses less resources, it is possible to move to a smaller, lower cost device.
- **Prototyping with Larger Devices** - This can reduce product development time.
- **Using Available Devices** - Current CPLD inventory can be used more efficiently.

2

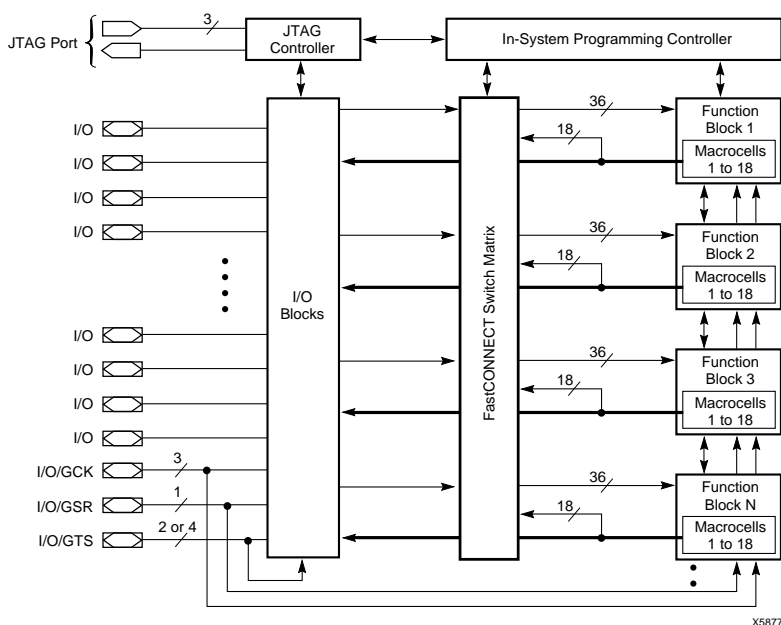


Figure 1: XC9500 Architecture

The XC9500 Architecture

Each device in the XC9500 family contains multiple uniform function blocks, connected by the FastCONNECT switch matrix, as shown in [Figure 1](#). This architecture guarantees 100% routability, even for fully populated devices. This highly flexible routing capability along with the advanced features of the Xilinx XACT fitter software provide highly reliable pin locking; this is a key requirement for reliable design migration. The XC9500 advanced pin-locking capability allows significant design changes without reworking PC boards.

XC9500 Packaging

For the whole XC9500 family, all device types in the same package have the same pinouts. For example, an XC9572 die in a PC84 package has the same pinouts as an XC95108 in that same package. Therefore, migrated designs will fit the same footprint, they will function almost identically, and PC boards will not need to be modified. Package availability is shown in [Table 1](#).

Design Migration Strategy

Moving an overcrowded design to a larger device is easy because the larger device will usually have an abundance of unused resources that make routing and pin-locking easier. Moving to a smaller device however, requires some previous planning. The following procedures show how to move designs into either a smaller or larger device.

Note1: These procedures only apply to different devices in the same package type. For example, upward design migration from an XC95108-PC84 design to an XC95144 cannot occur because the XC95144 is not available in the PC84 package. However it is possible to move an XC95108-PQ100 design into an XC95144 because the XC95144 is available in the PQ100 package.

Note2: These procedures and examples assume that the Xilinx XACT fitter, version 6.0.0 or later, is being used.

Migrating to a Larger Device

Moving a design to a larger device is very simple and straightforward. Usually this procedure is used when adding more functionality to an already full device or when using the larger devices in inventory.

1. Save the pinouts from the original, smaller, design. This is accomplished by selecting the “Lock the Pins” process in the XABEL-CPLD Project Navigator which creates a .GY2 (Pin-save) file that can later be used to specify pinouts.
2. Re-compile the original design, targeting the larger device, and using the original pinouts from the .GY2 file. To use the pinouts specified by the .GY2 file, highlight the “Fit Design” process in the XABEL-CPLD Project Navigator and click on “Properties”; “Fitter Options” is displayed. Select “Locked Pins” as the option for the “Pin Assignments” selection.
3. Review the Resource and Timing Reports generated by the fitter, to verify that the new design meets the specified pinout and timing requirements.

Migrating to a Smaller Device

Moving a design to a smaller device can also be accomplished, if the smaller device has enough I/O pins and macrocells to fit the design, and if the design is planned ahead. Usually this procedure is used if the smaller device is currently unavailable or if device resource requirements are likely to be reduced in the future.

Design migration into a smaller device is assured if the designer first targets the design to the smaller, harder to fit device, and allows the fitter to automatically assign pins. This will establish pinouts and timing based on the worst case device resource restrictions (the smaller device).

1. Compile the design, targeting the smaller device, without restricting pinouts. Save the pinouts into a Pin-Save file. This is accomplished by selecting the “Lock the Pins” process in the XABEL-CPLD Project Navigator which creates a .GY2 file that can later be used to specify pinouts. Verify that this design meets all of the functional and timing requirements.

Table 1: XC9500 Device Package Availability

Pins	XC9536	XC9572	XC95108	XC95144	XC95180	XC95216	XC95288	XC95432	XC95576
44	X								
84		X	X						
100		X	X	X					
160			X	X	X	X			
208					X	X	X		
352						X	X		
432								X	X

2. Re-compile the design, targeting the larger device, using the pinouts that were previously specified for the smaller device. To use the previous pinouts, highlight the “Fit Design” process in the XABEL-CPLD Project Navigator and click on “Properties”; “Fitter Options” is displayed. Select “Locked Pins” as the option for the “Pin Assignments” selection. Make sure this design meets all of the functional and timing requirements.
3. When both designs compile properly, and meet the design requirements, save the one targeted to the smaller device for later use.

Now, either the larger or smaller device can be used in the same socket on the PC board, and either design will function properly.

Design Migration Example

The following design example of a DRAM controller demonstrates the design migration procedure and shows how to compare the fitter reports to verify compatibility.

This design requires only 52 macrocells and therefore it will easily fit into an XC9572-7PC84. However, the design can also be targeted to an XC95108-7PC84 while maintaining both pinouts and performance.

This design, named “dram”, was first compiled for the XC9572, allowing pinouts to be assigned by the fitter. After the design compiled successfully, the automatically generated pinouts were verified and saved in a Pin-Save file for later use. Then the design was re-compiled, targeting the XC95108, using the saved pinouts from the previous XC9572 compilation. The timing for both designs was determined to be acceptable and both designs were saved for later use.

These two designs were compiled using the Xilinx XACT-CPLD V6.0.0 fitter which generates two reports:

- **dram.rpt** - This report shows a resource usage summary and shows where each signal is mapped in the device. Primarily, use this report to verify that the pinouts match for both devices and that both devices have adequate resources to fit the design.
- **dram.tim** - This report shows design timing. Primarily, use this report to verify that the design performance is adequate for both devices.

The following report examples show selected comparisons that demonstrate the functional compatibility of the two designs; these reports are abbreviated for clarity.

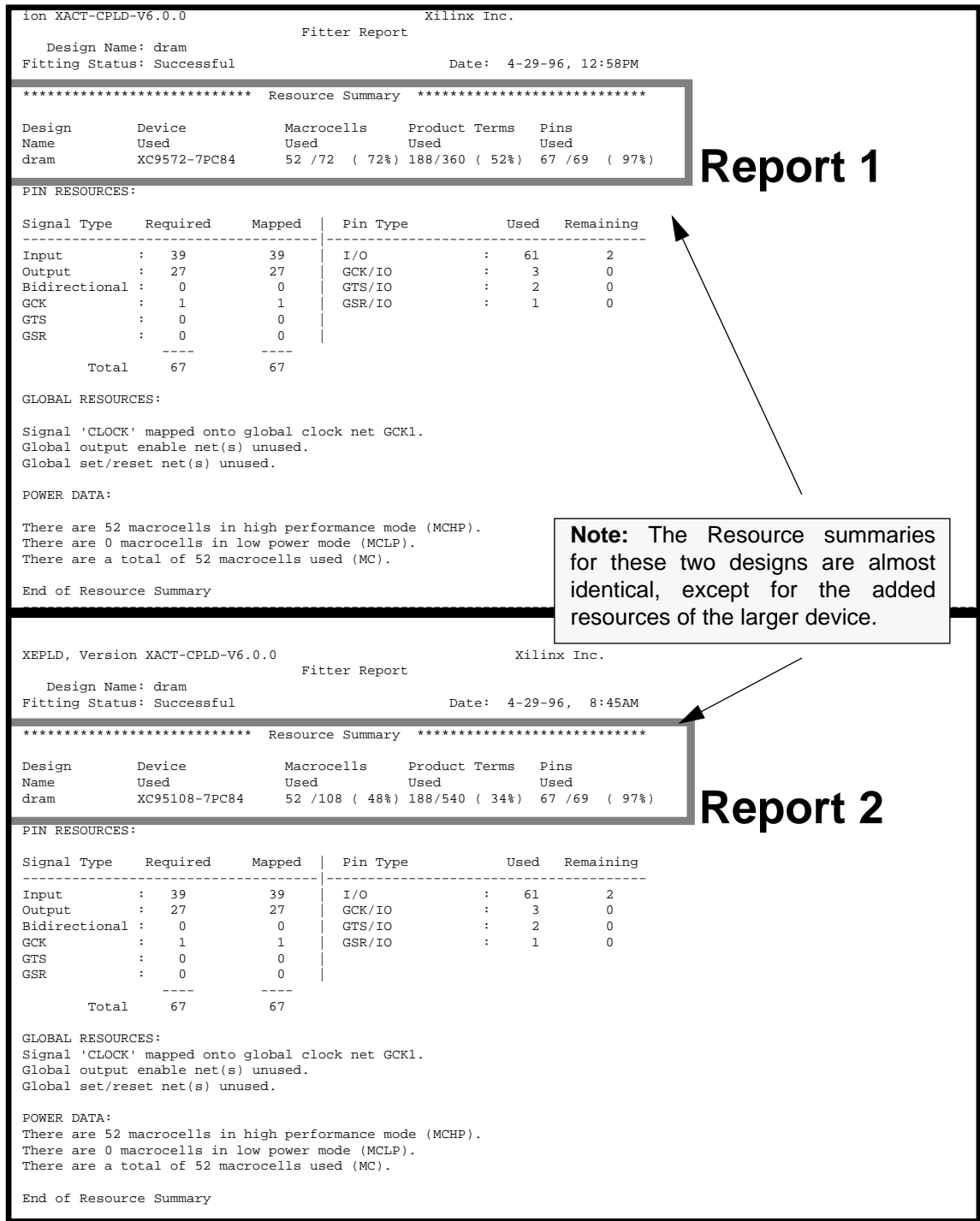
Resource Summary Report Examples

The reports illustrated in [Figure 2](#) show that both the XC9572 and the XC95108 have adequate resources to fit the DRAM controller design. The pinout diagrams for both designs were identical and are shown in [Figure 3](#).

Timing Report Examples

The reports illustrated in [Figure 4](#) and [Figure 5](#) show selected differences in various timing parameters for the two devices. From this report comparison it can be seen that the timing for these two designs is almost identical. If this were a real design, the designer would need to verify that these minor timing differences are not significant.

The timing report example shows some of the timing parameters for the XC9572 device; a sample of the XC95108 timing parameters that do not match identically, are highlighted separately. Those parameters that match exactly are not highlighted.



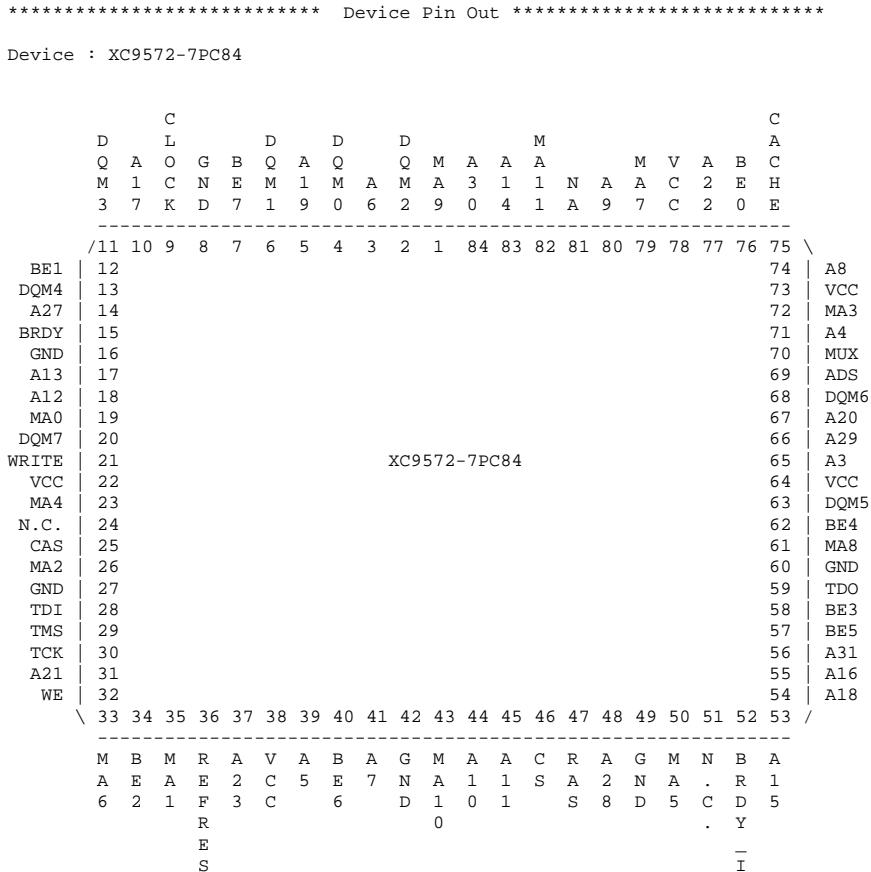


Figure 3: XC9572 Pinout Diagram (Identical for the XC95108)

Pinout Diagram

Figure 3 shows the pinout diagram for both reports, which were identical. Use this pinout diagram to verify that the pinouts are acceptable and identical for both designs.

Performance Summary:

Worst case Pad to Pad path delay : 7.5ns (1 macrocell levels)
 (Includes an external input margin of 0.0ns.)
 (Includes an external output margin of 0.0ns.)
 Pad 'A23' to Pad 'MA10'

Clock net 'CLOCK' path delays:

Worst case clock pad to output pad delay : 14.0ns (2 macrocell levels)
 (Includes an external input margin of 0.0ns.)
 (Includes an external output margin of 0.0ns.)
 Clock Pad 'CLOCK' to Output Pad 'MA10' (Fast Clock)

Clock to Setup path delay : 15.0ns (1 macrocell levels)
 Clock to Q, net 'P0_BANKB0.Q' to DFF Setup(D) at 'CS_MC.D' (Fast Clock)
 Target FF drives output net 'CS_MC'

Clock to Pad path delay : 11.5ns (2 macrocell levels)
 (Includes an external output margin of 0.0ns.)
 Clock to Q, net 'MUX_MC.Q' to Pad 'MA10' (Fast Clock)

Pad to Setup path delay : 17.5ns (1 macrocell levels)
 (Includes an external input margin of 0.0ns.)
 Pad 'CLOCK' to DFF Setup(D) at 'CS_MC.D' (Fast Clock)
 Target FF drives output net 'CS_MC'

Pad Ending at Clock Pin path delay : 2.5ns (0 macrocell levels)
 (Includes an external input margin of 0.0ns.)
 Clock Pad 'CLOCK' to DFF Clock Pin at 'FCLKIO_0' (Fast Clock)

Setup to Clock at the Pad : 15.0ns (1 macrocell levels)
 Data signal 'CLOCK' to DFF D input Pin at 'CS_MC.D'
 Clock pad 'CLOCK' to DFF Clock Pin at 'FCLKIO_0' (Fast Clock)

Minimum Clock Period: 15.0ns
 Maximum Internal Clock Speed: 66.6Mhz
 (Limited by Cycle Time)

Estimated Maximum External Clock Speed: 66.6Mhz
 (Limited by Cycle Time)

Figure 4: Timing Performance Summary (Identical for both devices)

Timing Performance Summary

Figure 4 shows the Timing Performance Summary. Use this part of the Timing Report to verify the overall timing for the design. This gives a quick overview of timing and reveals

any major timing differences. The two Performance Summaries are identical for both designs in this example, and therefore only one report is shown here for clarity.

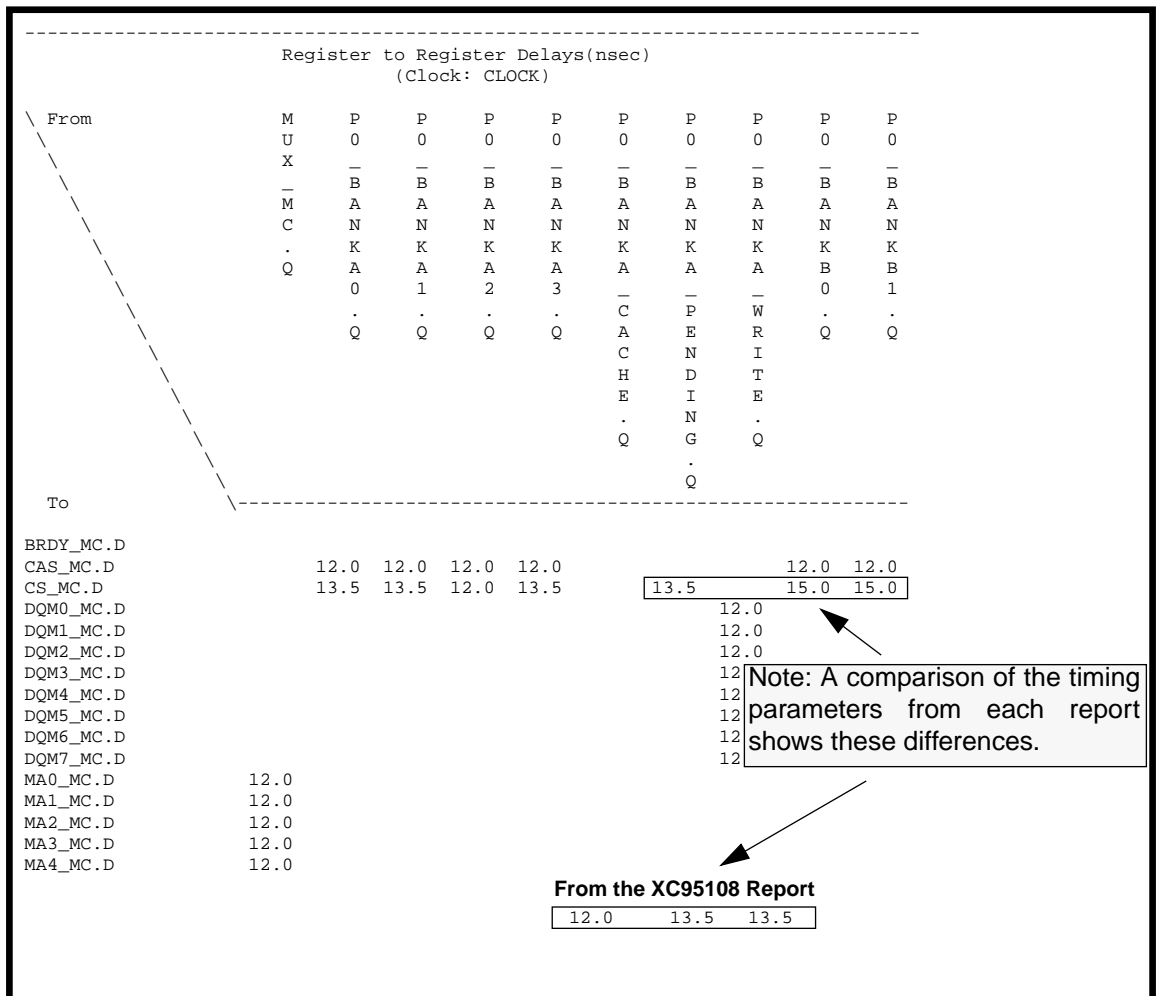


Figure 5: Timing Report Differences (XC9572 Report Shown Here)

Timing Report Differences

Figure 5 shows some of the timing differences from the two Timing Reports. Typically, the designer would compare each timing parameter from the two reports to be sure that all critical timing is within specification limits. In this example the timing difference is only 1.5ns for a register-to-register delay, which is well within the timing requirements for this design.

There were other timing differences between the two Timing Reports. However, the differences were very minor and are not illustrated here.

Conclusions

The flexibility of the XC9500 architecture, combined with the advanced features of the Xilinx fitter, give designers the capability to easily migrate designs into larger or smaller devices, while retaining the original pinouts. In addition, the timing differences between migrated designs are usually insignificant. This feature gives maximum design flexibility and allows designers to modify designs even after PC boards are laid out.

