Migrating from ACT3 to 42MX FPGAs

Table of Contents

Overview

This application note provides the information needed for migration of a design from the ACT3 families to the 42MX family. Based on architectural commonalities between these two families, the migration guideline is developed in this document. For example, both families share the proven two-module architecture consisting of combinatorial and sequential logic modules. This migration document outlines these commonalities as well as provides instruction on how to mitigate some of the architectural differences.

Comparison

Table 1 shows comparisons between the ACT3 and 42MX devices. The die and package specific migration path is shown in Table 2 on page 3. For a given ACT3 device, the recommended 42MX device is shown in the same colored grouping. In most cases, an equivalent package is available in the 42MX family; however, there are some packages in ACT3 that do not exist for the recommended 42MX device. In such scenarios, you must change to a different package or consult Microsemi sales for development of a new package. Table 1 highlights (in yellow) these differences. Several options will be discussed in this document.

Table 1 • **Comparison Table**

Devices	**A1415/ A14v15**	**A42MX09**	**A1425/ A14v25**	**A1440/ A14v40**	**A42MX16**	**A1460/ A14v60**	**A42MX24**	**A14100/ A14v100**	**42MX36**
System Gates	3,750	14,000	6,250	10,000	24,000	15,000	36,000	25,000	54,000
Logic modules	200	684	310	564	1232	848	1866	1377	2414
Sequential	104	348	160	288	624	432	954	697	1230
Combinatorial	96	336	150	276	608	416	912	680	1184
Dedicated Flip-flops	264	348	360	568	624	768	954	1153	1230
Clocks	3	2	3	3	2	3	2	3	6
User I/O (Maximum)	80	104	100	140	140	168	176	228	202
Packages (by pin counts)									
CPGA	PG100	N/A	PG133	PG175	N/A	PG207	N/A	PG257	N/A
PLCC	84	84	84	84	84	N/A	84	N/A	N/A
PQFP	100	100,160	100,160	160	100,160, 208	160,208	160,208	N/A	208,240
RQFP	N/A	N/A	N/A	N/A	NA	N/A	N/A	208	N/A
VQFP	100	100	100	100	100	N/A	N/A	N/A	N/A
TQFP	N/A	176	N/A	176	176	TQ160	TQ160	–	–
PBGA	N/A	N/A	N/A	N/A	N/A	225	N/A	313	272
CQFP	N/A	N/A	132	N/A	N/A	196	N/A	256	208,256, 272

Note: "v" stands 3.3 V VCC (core) support

Table 2 • **Die/Package Migration list**

Legacy Die / Package	Suggested Migration	Note
A1415/A14v15	**A42MX09**	–
PG100	*No compatible package	–
PL84	PL84	–
PQ100	PQ100	–
VQ100	VQ100	–
A1425/A14v25	**A42MX16**	–
PG133	*No compatible package	–
PL84	PL84	–
PQ100	PQ100	–
PQ160	PQ160	–
VQ100	VQ100	–
CQ132	*No compatible package	–
A1440/A14v40	**A42MX16**	–
PG175	*No compatible package	–
PL84	PL84	–
PQ160	PQ160	–
VQ100	VQ100	–
TQ176	TQ176	–
A1460/A14v60	**A42MX24**	**–**
PG207	*No compatible package	–
PQ160	PQ160	–
PQ208	PQ208	–
TQ176	TQ176	–
BG225	*No compatible package	BG272 can be explored
CQ196	*No compatible package	–
A14100/A14v100	**A42MX36**	**–**
PG257	*No compatible package	–
RQ208	*No compatible package	PQ208 can be explored as the package footprint is the same
BG313	*No compatible package	–
CQ256	CQ256	–

Notes:

"v" stands 3.3 V VCC (core) support

** For more information contact Sales*

Pin-to-Pin Comparison

Power Supply Pin Comparison

The supply voltage pins of the 42MX family are fully compatible with the ACT3 devices. There are several new pins in the 42MX devices that did not exist in the ACT3 devices:

- VCCA and VCCI: In the 42MX family, the supply voltage for the array is VCCA, and the supply voltage for I/Os is VCCI. Both the array and I/Os are supplied by VCC in the ACT3 family.
- LP: Low Power Mode pin. In this mode, all I/Os are tristated, all input buffers are turned off, and the core of the device is turned off. This feature is particularly useful for battery-operated systems where battery life is a primary concern. This pin must be tied low unless you make board level provisions to utilize this function.
- NC: No Connection. This pin is not connected to circuitry within the device. These pins can be driven to any voltage or can be left floating with no effect on the operation of the device.

Table 3 lists the power supply recommendations for the new pins in the 42MX devices.

Table 3 • **Power Supply Pin Recommendation**

ACT3 Pin	42MX Pin	Recommendation
VCC	VCCA	Connect to board VCC supply
VCC	VCCI	Connect to board VCC supply
VCC	NC	Connect to board VCC supply
GND	LP/GND	Connect to GND

I/O Comparison

The ACT3 devices have a slightly different I/O module than 42MX devices. Each ACT3 I/O module contains two additional pins, IOCLK and IOPCL, which work as clock and preset/clear inputs for the two I/O flip-flops in the I/O module. The 42MX devices do not have I/O flip-flops. To secure an I/O timing, regular core flip-flop can be used and hand placed near the I/O buffer in the 42MX devices.

Review the Antifuse Macro Library Guide for Software v9.0SP1 for details on the macro definitions.

A list of macros that are not available in 42MX library can be viewed in the "Appendix 2".

Clock Comparison

ACT3 devices offer one HCLK (hardwired) dedicated clock network that directly drives the clock input of all array/core registers, which the 42MX devices do not offer. Both ACT3 and 42MX devices offer two routed clock networks, CLKA and CLKB. Thus, the number of total dedicated clocks in 42MX devices, except 42MX36, is less than ACT3 devices. Careful design consideration should be made with dedicated clocks. Regular I/Os can be used as clocking sources where the timing is not so critical. The 42MX36 devices contain quadrant clocks, which partially mitigates this issue as these clocks can be driven across one fourth of the device.

Special Pins Comparison

MODE, PROBE, SDI, SDO, TDI, TDO, and JTAG pins are fully compatible in terms of location, default configuration, and functionality.

Power Supply Voltages

Compatible devices run on the same voltage level of 5 V or 3.3 V. The 42MX devices can also operate in mixed 5 V / 3.3 V systems (Table 4).

Table 4 • **Power Supply Voltage Comparison**

Device	VCC	VCCA	VCCI	Maximum Input Tolerance	Nominal Output Voltage
ACT3	5.0 V	–	–	5.5 V	5.0 V
	3.3 V	–	–	3.6 V	3.3 V
42MX	–	5.0 V	5.0 V	5.5 V	5.0 V
	–	3.3 V	3.3 V	3.6 V	3.3 V
	–	5.0 V	3.3 V	5.5 V	3.3 V

Power-Up Recommendation

When powering up 42MX in mixed-voltage mode, VCCA must be greater than or equal to VCCI throughout the power-up sequence. If VCCI exceeds VCCA during power-up, either the I/O input protection junction at the I/Os will be forward-biased or the I/Os will be at logical HIGH, resulting in ICC rising to high levels. Microsemi strongly recommends this power-up sequence to ensure the device's functionality. For the purpose of migrating to 42MX, this will not be an issue, since VCCA and VCCI will be connected to the same board VCC supply.

Configuration of Unused I/Os

The same configuration applies to both families (Table 5).

Table 5 • **Unused I/Os for All families**

Legacy Device	Comparable Device	Configuration
A1415, A1425, A1440	A42MX09, A42MX16	Pulled LOW
A1460, A14100	A42MX24, A42MX36	Tristated

Unused I/Os are automatically configured by the Libero® IDE/Designer software. In all cases, Microsemi recommends you tie all unused 42MX I/Os to LOW on the board. For more information, refer to the Knowledge Base document, "Default Settings for Unused I/O and Clocks".

Design Migration Procedure

Due to some architectural differences manual migration procedure needs to be adopted. There are certain logic cells in ACT3 which are not present in the 42MX library cells, and an equivalent functional block needs to be created to mitigate this (refer to "Appendix 1"). A list of ACT3 macro cells (that is, logic cells) that are not available in the 42MX library can be viewed in "Appendix 2" to help in the migration procedure. Here is the suggested manual migration flow:

Step 1: Opening an Existing ACT3 File in Designer

File Required: *Original.adb*

This step assumes that the ACT3 ADB file has the complete information up to Layout. If only the Netlist (ADL/EDN) and the other constraint source files are available, rather than an ADB, skip to "Step 3: Creating a New 42MX File in Designer" section on page 9. The ADL/EDN Netlist file needs to be converted to [1]VHDL Netlist (import the ADL/EDN file to a ACT3 project in Designer and export the VHDL netlist from File>Export>Netlist).

1. *You can export verilog netlist and create an equivalent functional block in verilog for non supported library cells in 42MX.*

Open the Designer software and select **Open Existing Design** to open the ADB file. When Designer is opened, the Compile and Layout icons should be green (Figure 1).

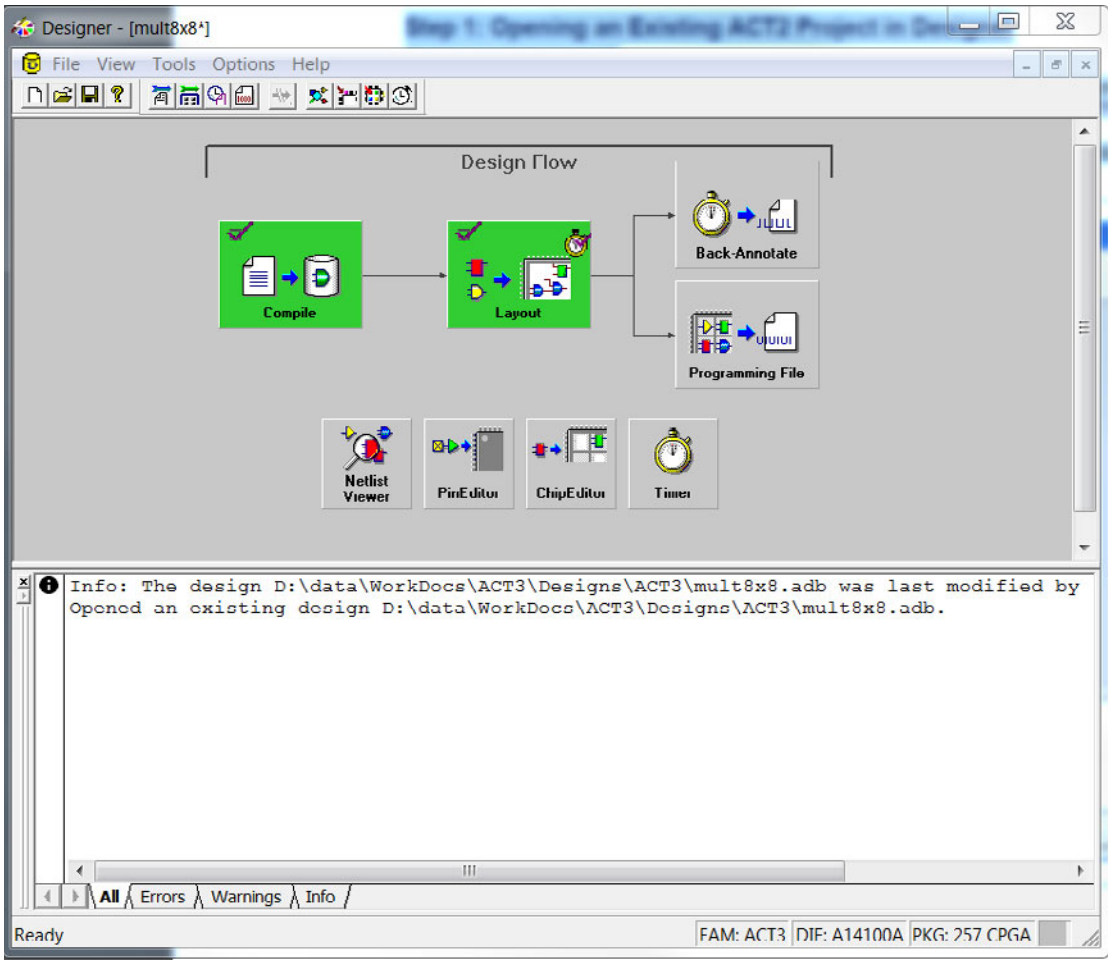

***Figure 1* • Opening a Completed Design Database (adb)**

Step 2: Exporting the VHDL Netlist, Pinout List, and Timing file from Designer

Select **File > Export > Netlist Files** to export the VHDL Netlist (Figure 2).

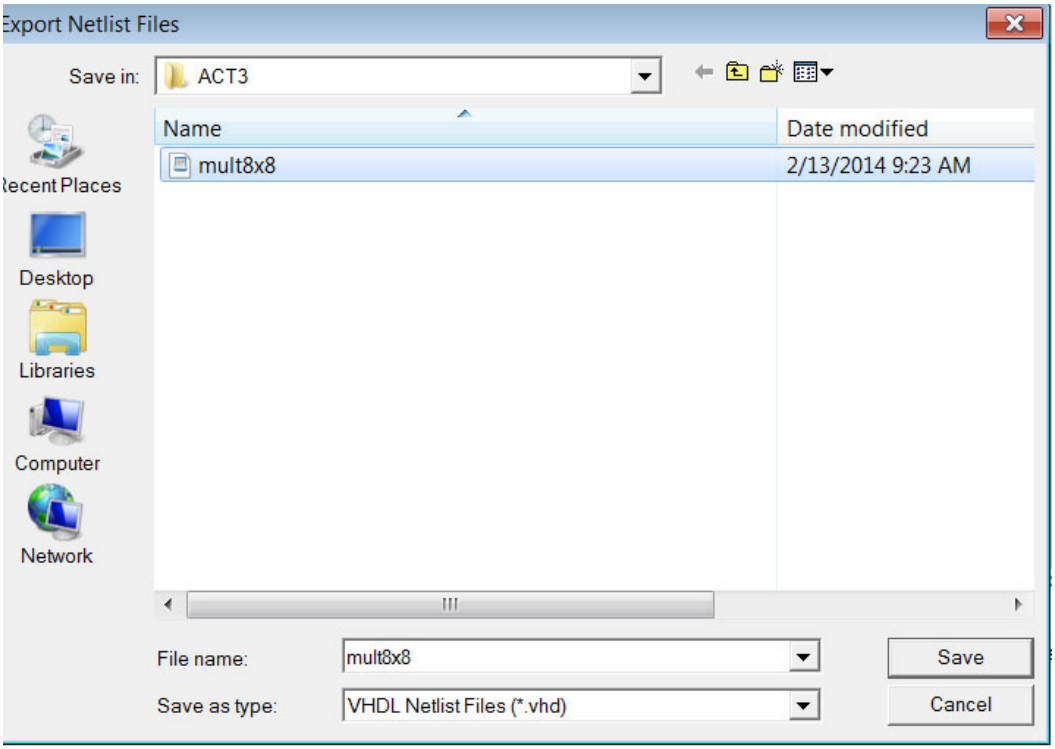

***Figure 2* • Exporting VHDL Netlist File**

Choose **File > Export >Constraint files** to export the pin list (*.pin) and timing files (*.dcf) and (*.crt) from Designer (Figure 3).

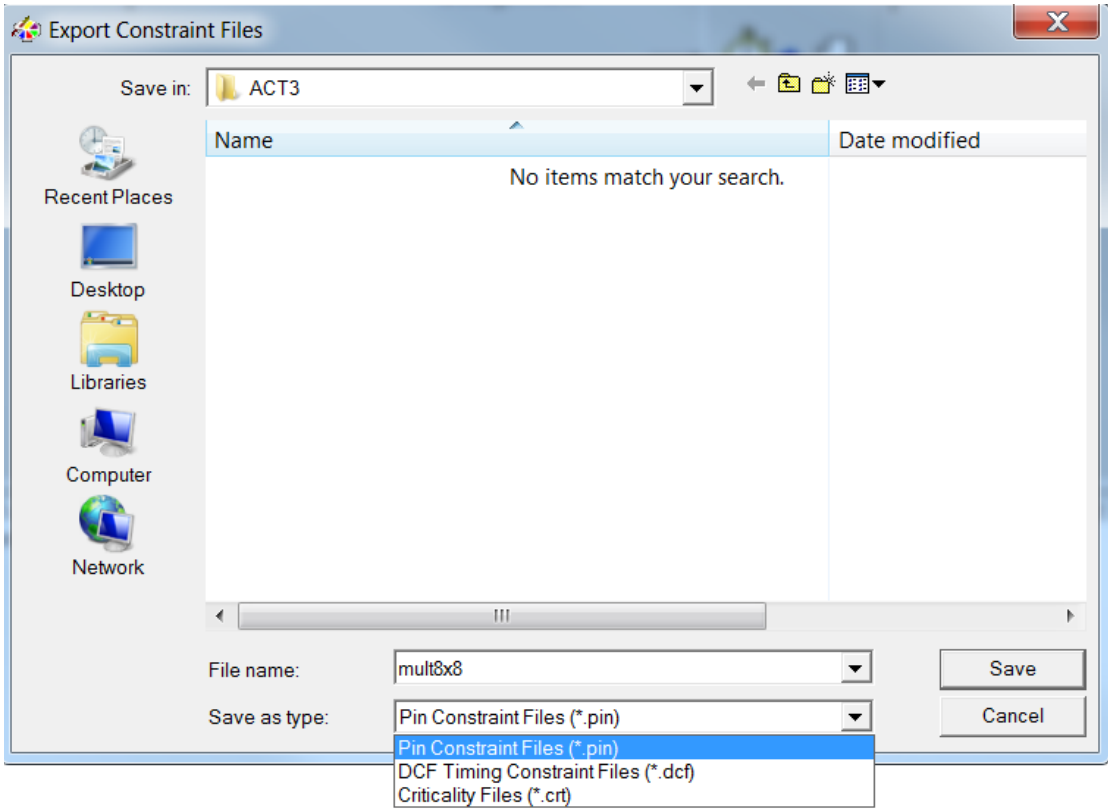

***Figure 3* • Exporting Constraint Files**

Step 3: Creating a New 42MX File in Designer

In Designer, choose **File > New**. Enter the **design name** and select the family **42MX**, as shown in Figure 4.

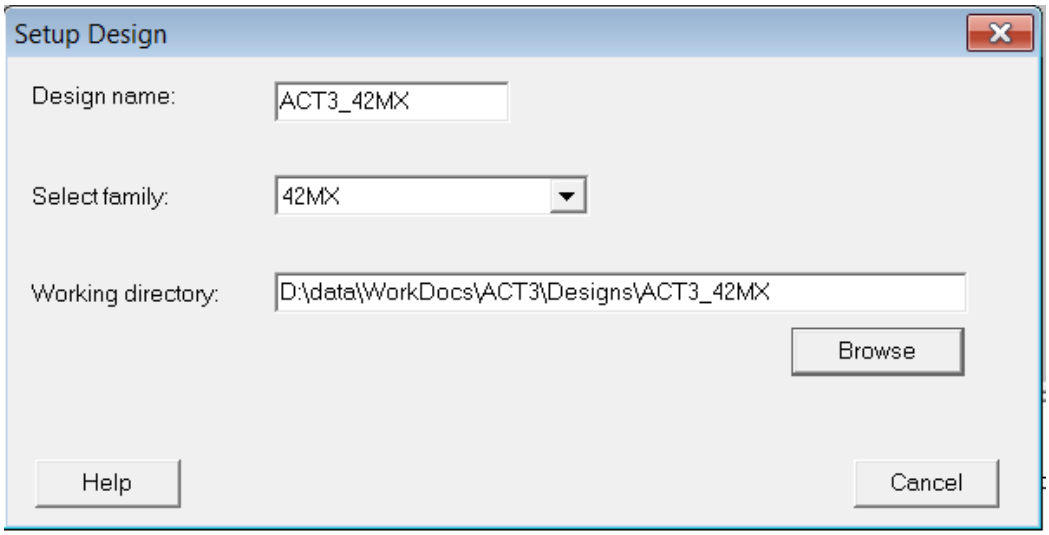

***Figure 4* • Setup 42MX Design**

Step 4: Importing Netlist and Constraint files Back into the New 42MX Design

Since some ACT3 library cells are not present in the 42MX library, first import only the Netlist and then go through the compile step. Incompatible cells will be printed out in the Designer log window as follows:

```
Warning: DFM8A is not an Actel cell defined in the cell library.
Warning: HCLKBUF is not an Actel cell defined in the cell library.
```

Create the equivalent library files to continue the migration process. Refer to "Appendix 1" for details.

Choose **File > Import Source Files**. Add the missing macro library file (**ACT3_Macros.vhd**) and component declaration package (**comps.vhd**) along with **VHDL Netlist**, **PIN**, **CRT** files of ACT3 into the new 42MX design file, as shown in Figure 5 .

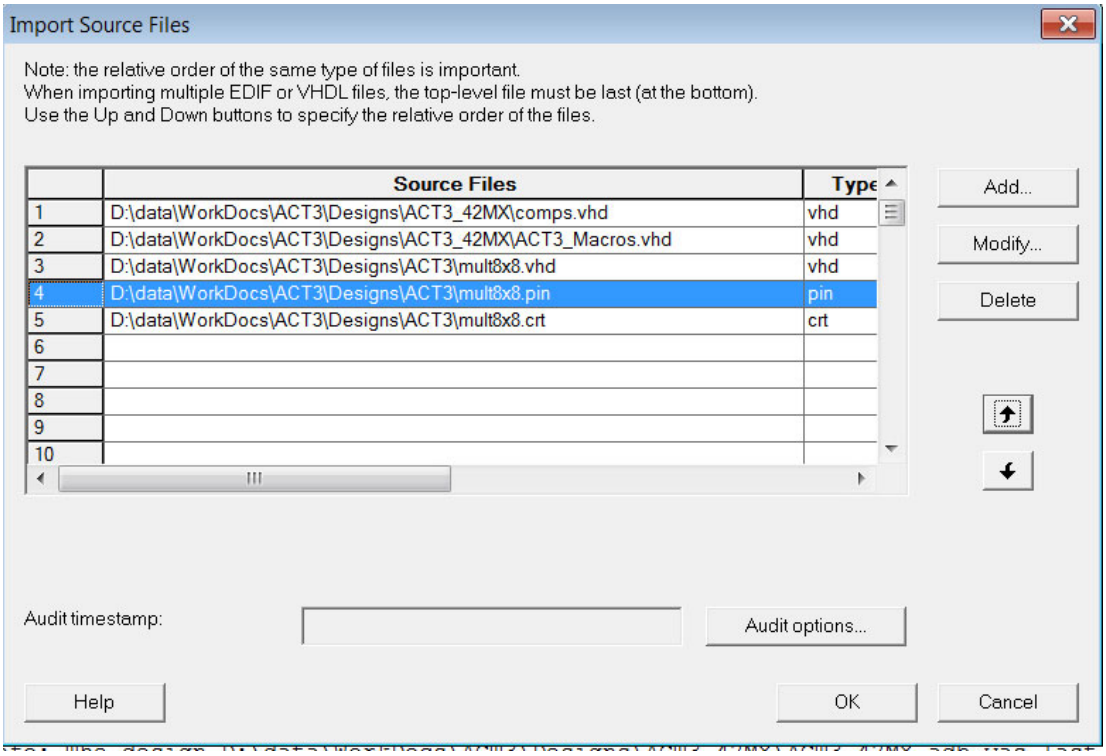

Figure 5 • **Importing Source Files in 42MX Design**

Refer to the "Appendix 1" for details on how to create_ACT3_Macros.vhd and comps.vhd.

Step 5: Follow the Design Flow

Click the **Compile button** and the Device Selection Wizard will open. Choose the compatible 42MXDie/package (refer to Table 1 and Table 2), as shown in Figure 6.

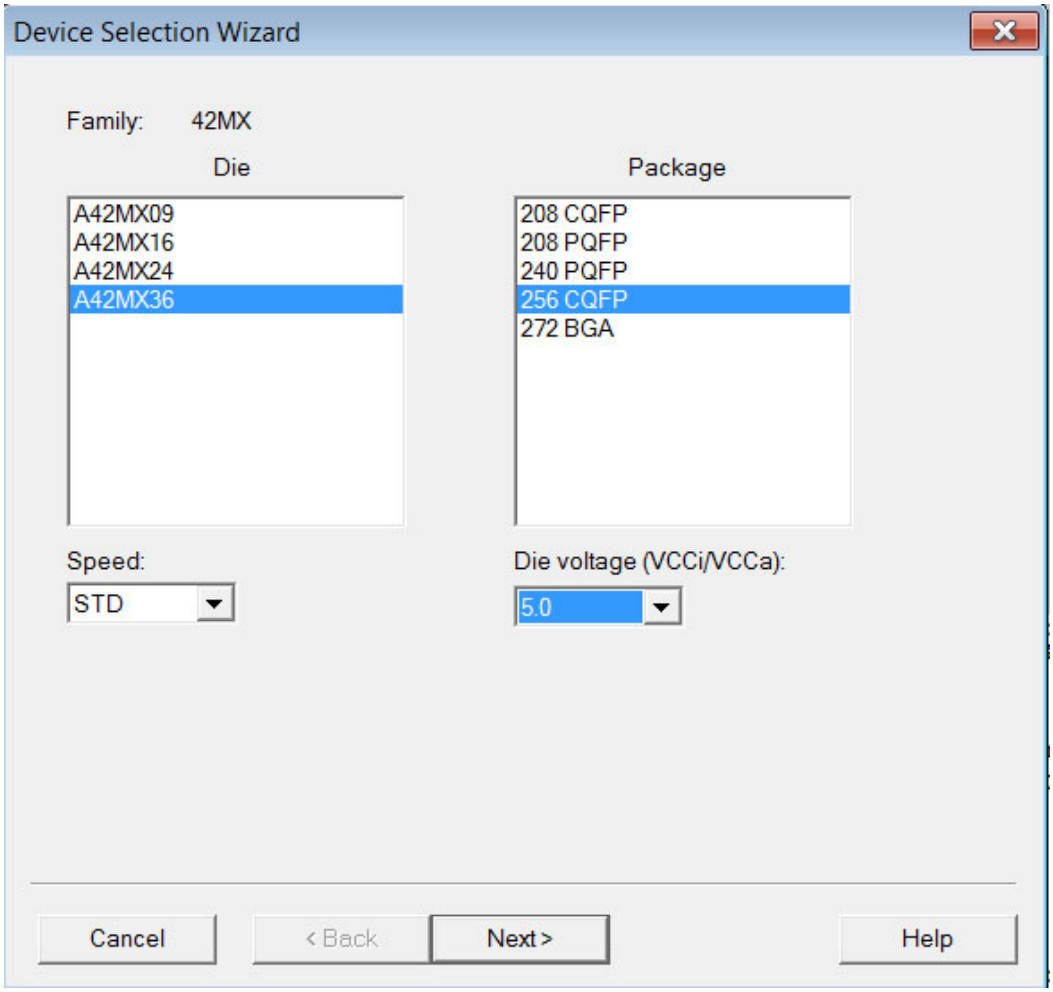

***Figure 6* • Device Selection Wizard**

Set the **identical operating conditions** (same as used for the original ACT3 design) and carefully choose the **device speed grade** for the new 42MX design. Examples of the operating conditions are VCCI/VCCA, Restrict-Probe-pins, Junction temperature, and Voltage range.

Complete the Compile action and import the timing file (**dcf**) into the design. Choose **File > Import Auxiliary File** to add the timing file into the design, as shown in Figure 7.

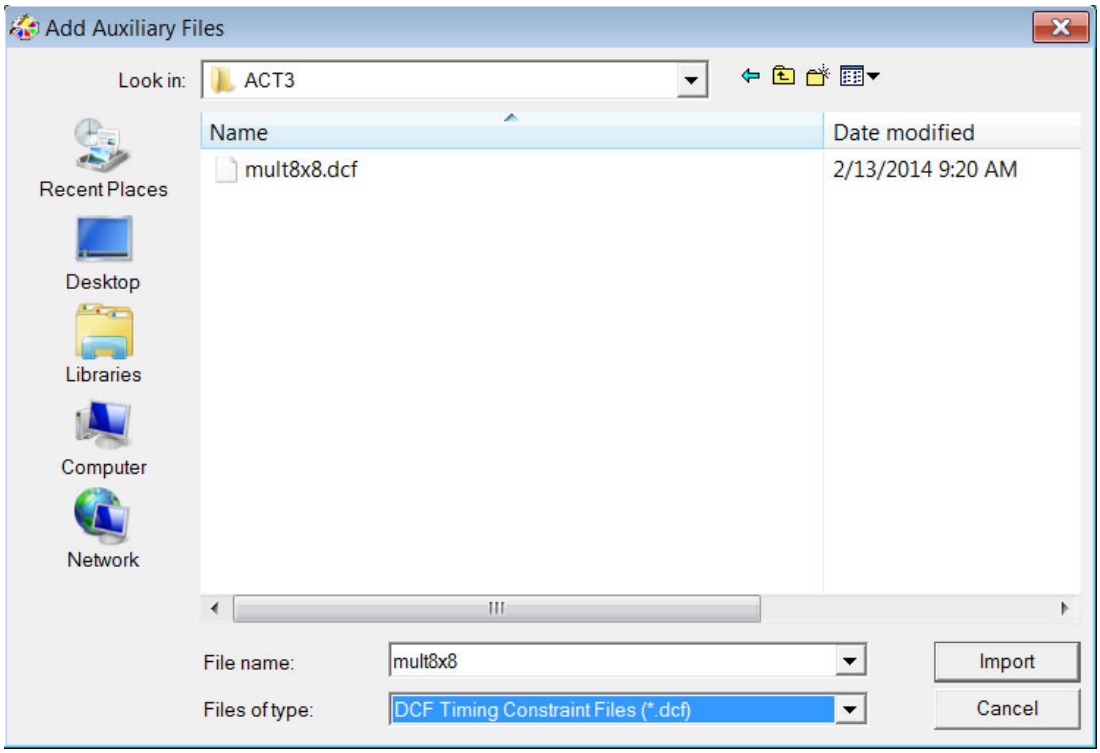

***Figure 7* • Add Timing File (dcf) into the Project**

Click on **Layout** to complete the place and route of the design. Open **Pin Editor** to verify the pin outs.

Timing Concerns

The 42MX family is generally faster than the ACT3 family. Therefore, Microsemi recommends that you perform a new timing analysis and pay attention to the hold time, cross clock domain paths, clock-to-out, and multicycle paths, although it is expected that the migrated design will have the same timing—if the ACT3 timing constraint file (*.dcf) was exported. Refer to the "Static Timing Analysis Using Designer's Timer" application note for more information on performing the timing analysis using the Microsemi Timer tool. Verify the potential simultaneously switching outputs by checking whether various adjacent outputs have enough timing differences (staggered timing) to avoid negative effects. Refer to Simultaneous Switching Noise and Signal Integrity application note for details. The 42MX I/Os have different slew rate than the ACT3 devices. Run a board level signal integrity analysis before finalizing the board design. Refer to the Using Schmitt Triggers for Low Slew-Rate Input application note for details.

If the original design was created with Designer v3.1 or older, it requires the Object Store server to be running in order to convert the design to be compatible to the newer version of Designer. This can be done if Designer v3.x is available by following the installation and setup instruction manual for that release of Designer. If the Object Store server is not available, contact Technical Support by email at soc_tech@microsemi.com and they will try converting the design.

Programming

Programming Software

Programming files are not compatible between the two families. A new programming file (AFM) must be generated from the migrated design.

Programming Hardware

Silicon Sculptor 3: Uses the same module for both families.

Conclusion

The 42MX family shares many architectural features and the library of basic elements with the ACT3 families—with the exception of some library cells (refer to "Appendix 1"), and offers higher speed and special functionalities. Understanding the differences between the two families makes a seamless migration from the ACT3 families to the 42MX family possible.

Related Documents

Datasheets

ACT3 Families

40MX and 42MX FPGA Families

Application Notes

AC112: Using ACT 3 Family I/O Macros App Note

42 MX Family Devices Power-Up Behavior

Static Timing Analysis Using Designer's Timer

Other Documents

Default Settings for Unused I/O and Clocks

Libero IDE Design Flow

Appendix 1

ACT3 and 42MX library cell difference:

Some of the ACT3 library cells are not supported in the 42MX library. A VHDL library needs to be created, which will provide the equivalent logic of the ACT3 library cells for the 42MX design. Create two VHDL files, a library file that provides the equivalent functionality by using library cells from the 42MX library (ACT3_macros.vhd) and a component declaration file (comps.vhd).

The following shows the difference in the library of these two families. A complete list of ACT3 library cells that are not supported in 42MX can be viewed in "Appendix 2".

1. **I/Os:** ACT3 I/Os are different than MX. Each I/O has IOCLK and IOPCL for I/O flip-flop. Create the equivalent I/O soft macro for 42MX. Use the sequential cell and CLK from 42MX to replace IOCLK and IOPCL (this can be a regular I/Os).
2. **CLOCK:** ACT3 has HCLK and does not exist in 42MX. This could be a very big hit in achieving the clock speed. 42MX has QCLK but this is limited to quad only. These QCLKs can be used as an alternative to IOCLK and IOPCL but again, they are limited in number (only four QCLK). The following example shows an equivalent RTL code of a ACT3 HCLK that is not present in the 42MX library:

```
library ieee;
use ieee.std_logic_1164.all;
library work;
use work.components.all;

entity HCLKBUF is
    port(
   PAD   : in  std_logic;
   Y     : out std_logic);
end HCLKBUF ;

architecture rtl of HCLKBUF is
    begin
   U1 : CLKBUF
      port map(PAD => PAD,
       Y => Y);
end rtl;
```

3. Core logic: Study the truth table of the ACT3 macro library guide and create equivalent logic cells for the 42MX library. The following example shows an equivalent RTL code of a ACT3 library cell "DFM8A" that is not present in 42MX library:

```
library ieee;
use ieee.std_logic_1164.all;
library work;
use work.components.all;

entity DFM8A is
    port(
   D0    : in  std_logic;
   D1    : in  std_logic;
   D2    : in  std_logic;
   D3    : in  std_logic;
   S00   : in  std_logic;
   S01   : in  std_logic;
   S10   : in  std_logic;
   S11   : in  std_logic;
   CLK   : in  std_logic;
   CLR   : in  std_logic;
   Q     : out std_logic);
end DFM8A ;

architecture rtl of DFM8A is
```

```
SIGNAL mx1_sel : std_logic;
SIGNAL mx2_sel : std_logic;
SIGNAL mx3_sel : std_logic;
SIGNAL mx3_A   : std_logic;
SIGNAL mx3_B   : std_logic;
SIGNAL reg_in  : std_logic;

    begin

   -- Select logic for mux1
   A1 : AND2
   port map(A => S00,
       B => S01,
       Y => mx1_sel);

   -- Select logic for mux2
   A2 : AND2
   port map(A => S00,
       B => S01,
       Y => mx2_sel);

   -- Select logic for mux3
   O1 : OR2
   port map(A => S10,
       B => S11,
       Y => mx3_sel);

   M1 : MX2
   port map(A => D0,
       B => D1,
       S => mx1_sel,
       Y => mx3_A);

   M2 : MX2
   port map(A => D2,
       B => D3,
       S => mx2_sel,
       Y => mx3_B);

   M3 : MX2
   port map(A => mx3_A,
       B => mx3_B,
       S => mx3_sel,
       Y => reg_in);

   R1 : DFC1B
   port map(D   => reg_in,
       CLK => CLK,
       CLR => CLR,
       Q   => Q);
end rtl;
```

4. Example of component declaration file (comps.vhd):

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

package COMPONENTS is


COMPONENT CLKBUF
   port(PAD: in    STD_ULOGIC;
        Y: out    STD_ULOGIC);
```

```
END COMPONENT;

COMPONENT AND2
    port(
                A          : in    STD_ULOGIC;
                B          : in    STD_ULOGIC;
               Y                : out    STD_ULOGIC);
END COMPONENT;

COMPONENT OR2
    port(
                A          : in    STD_ULOGIC;
                B          : in    STD_ULOGIC;
               Y                : out    STD_ULOGIC);
END COMPONENT;

COMPONENT MX2
    port(
                A          : in    STD_ULOGIC;
                S          : in    STD_ULOGIC;
                B          : in    STD_ULOGIC;
               Y                : out    STD_ULOGIC);
END COMPONENT;

COMPONENT DFC1B
    port(
   D     : in  std_logic;
   CLK   : in  std_logic;
   CLR   : in  std_logic;
   Q     : out std_logic);
END COMPONENT;

END COMPONENTS;
```

Appendix 2

Table 6 • **Macro List**

module BBHSA(D,E,IDE,PAD,Y);
inout PAD;
input D, E, IDE;
output Y;
endmodule
module BBLSA(D,E,IDE,PAD,Y);
inout PAD;
input D, E, IDE;
output Y;
endmodule
module BBUFTH(D, E, PAD, Y);
inout PAD;
input D, E;
output Y;
endmodule
module BBUFTL(D, E, PAD, Y);
inout PAD;
input D, E;
output Y;
endmodule
module BIECTH(CLK, D, E, IOPCL, IDE, PAD, Q);
inout PAD;
input CLK, D, E, IOPCL, IDE;
output Q;
endmodule
module BIECTL(CLK, D, E, IOPCL, IDE, PAD, Q);
inout PAD;
input CLK, D, E, IOPCL, IDE;
output Q;
endmodule
module BIEPTH (CLK, D, E, IOPCL, IDE, PAD, Q);
inout PAD;
input CLK, D, E, IOPCL, IDE;
output Q;
endmodule

Table 6 • **Macro List**

module BIEPTL (CLK, D, E, IOPCL, IDE, PAD, Q);
inout PAD;
input CLK, D, E, IOPCL, IDE;
output Q;
endmodule
module BRECTH(CLK, D, E, IOPCL, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module BRECTL(CLK, D, E, IOPCL, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module BREPTH(CLK, D, E, IOPCL, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module BREPTL(CLK, D, E, IOPCL, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module DECETH(CLK, D, E, IDE, IOPCL, ODE, PAD, Q);
inout PAD;
input CLK, D, E, IDE, IOPCL, ODE;
output Q;
endmodule
module DECETL(CLK, D, E, IDE, IOPCL, ODE, PAD, Q);
inout PAD;
input CLK, D, E, IDE, IOPCL, ODE;
output Q;
endmodule
module DEPETH(CLK, D, E, IDE, IOPCL, ODE, PAD, Q);
inout PAD;

Table 6 • **Macro List**

input CLK, D, E, IDE, IOPCL, ODE;
output Q;
endmodule
module DEPETL(CLK, D, E, IDE, IOPCL, ODE, PAD, Q);
inout PAD;
input CLK, D, E, IDE, IOPCL, ODE;
output Q;
endmodule
module DFM8A(D0, D1, D2, D3, S00, S01, S10, S11, CLK, CLR, Q);
input D0, D1, D2, D3, S00, S01, S10, S11, CLR, CLK;
output Q;
endmodule
module DFM8B (D0, D1, D2, D3, S00, S01, S10, S11, CLK, CLR, Q);
input D0, D1, D2, D3, S00, S01, S10, S11, CLR, CLK;
output Q;
endmodule
module DLM8A(D0, D1, D2, D3, S00, S01, S10, S11, G, CLR, Q);
input D0, D1, D2, D3, S00, S01, S10, S11, G, CLR;
output Q;
endmodule
module DLM8B(D0, D1, D2, D3, S00, S01, S10, S11, G, CLR, Q);
input D0, D1, D2, D3, S00, S01, S10, S11, G, CLR;
output Q;
endmodule
module FECTH(CLK, D, E, IOPCL, ODE, PAD, Y);
output PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module FECTL(CLK, D, E, IOPCL, ODE, PAD, Y);
output PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module FECTMH(CLK, D, E, IOPCL, M, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, M, ODE;

Table 6 • **Macro List**

output Y;
endmodule
module FECTML(CLK, D, E, IOPCL, M, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, M, ODE;
output Y;
endmodule
module FEPTH (CLK, D, E, IOPCL, ODE, PAD, Y);
output PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module FEPTL (CLK, D, E, IOPCL, ODE, PAD, Y);
output PAD;
input CLK, D, E, IOPCL, ODE;
output Y;
endmodule
module FEPTMH(CLK, D, E, IOPCL, M, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, M, ODE;
output Y;
endmodule
module FEPTML(CLK, D, E, IOPCL, M, ODE, PAD, Y);
inout PAD;
input CLK, D, E, IOPCL, M, ODE;
output Y;
endmodule
module HCLKBUF(PAD, Y);
input PAD;
output Y;
endmodule
module IBUF(PAD, Y);
input PAD;
output Y;
endmodule
module IOCLKBUF(PAD, Y);
input PAD;

Table 6 • **Macro List**

output Y;
endmodule
module IODFE(D, E, CLK, Q);
input D, E, CLK;
output Q;
endmodule
module IODFEC(D, E, CLK, IOPCL, Q);
input D, E, CLK, IOPCL;
output Q;
endmodule
module IODFEP(D, E, CLK, IOPCL, Q);
input D, E, CLK, IOPCL;
output Q;
endmodule
module IOPCLBUF(PAD, Y);
input PAD;
output Y;
endmodule
module IREC(CLK, IDE, IOPCL, PAD, Q);
input CLK, IDE, IOPCL, PAD;
output Q;
endmodule
module IREP(CLK, IDE, IOPCL, PAD, Q);
input CLK, IDE, IOPCL, PAD;
output Q;
endmodule
module OBUFTH(D, E, PAD);
input D, E;
output PAD;
endmodule
module OBUFTL(D, E, PAD);
input D, E;
output PAD;
module ORECTH(CLK, D, E, IOPCL, ODE, PAD);
input CLK, D, E, IOPCL, ODE;
output PAD;
endmodule

Table 6 • **Macro List**

module ORECTL(CLK, D, E, IOPCL, ODE, PAD);
input CLK, D, E, IOPCL, ODE;
output PAD;
endmodule
module OREPTH (CLK, D, E, IOPCL, ODE, PAD);
input CLK, D, E, IOPCL, ODE;
output PAD;
endmodule
module OREPTL (CLK, D, E, IOPCL, ODE, PAD);
input CLK, D, E, IOPCL, ODE;
output PAD;
endmodule
module A3IOMOD1(Y, PAD, D, E, IDE, SLEW);
inout PAD;
input D, E, IDE, SLEW;
output Y;
endmodule
module A3IOMOD2(Y, PAD, CLK, D, E, IDE, IEN, ODE, OTB, PCL, SLEW, SRS);
inout PAD;
input CLK, D, E, IDE, IEN, ODE, OTB, PCL, SLEW, SRS;
output Y;
endmodule
module A3IOMOD3(Y, PAD, CLK, D, E, IDE, IEN, ODE, OTB, SLEW, SRS);
inout PAD;
input CLK, D, E, IDE, IEN, ODE, OTB, SLEW, SRS;
output Y;
endmodule

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at www.microsemi.com.