

# ATMEL CPLD DEVELOPMENT KIT

## Tutorial 1: Complete CPLD Design Flow

This Tutorial will guide you through a complete CPLD design cycle with Logic Doubling™ from design entry, logic synthesis, device fitting, in-system programming, and finally verifying the design on the Atmel CPLD Demo Board. To complete this tutorial, ProChip Designer V3.0 or later and V1.8.4 or later fitters are required.

**Step I: Create a Project using ProChip Designer's New Project Wizard**

**Step II: Add a CUPL Design File**

**Step III: Compile the CUPL Design**

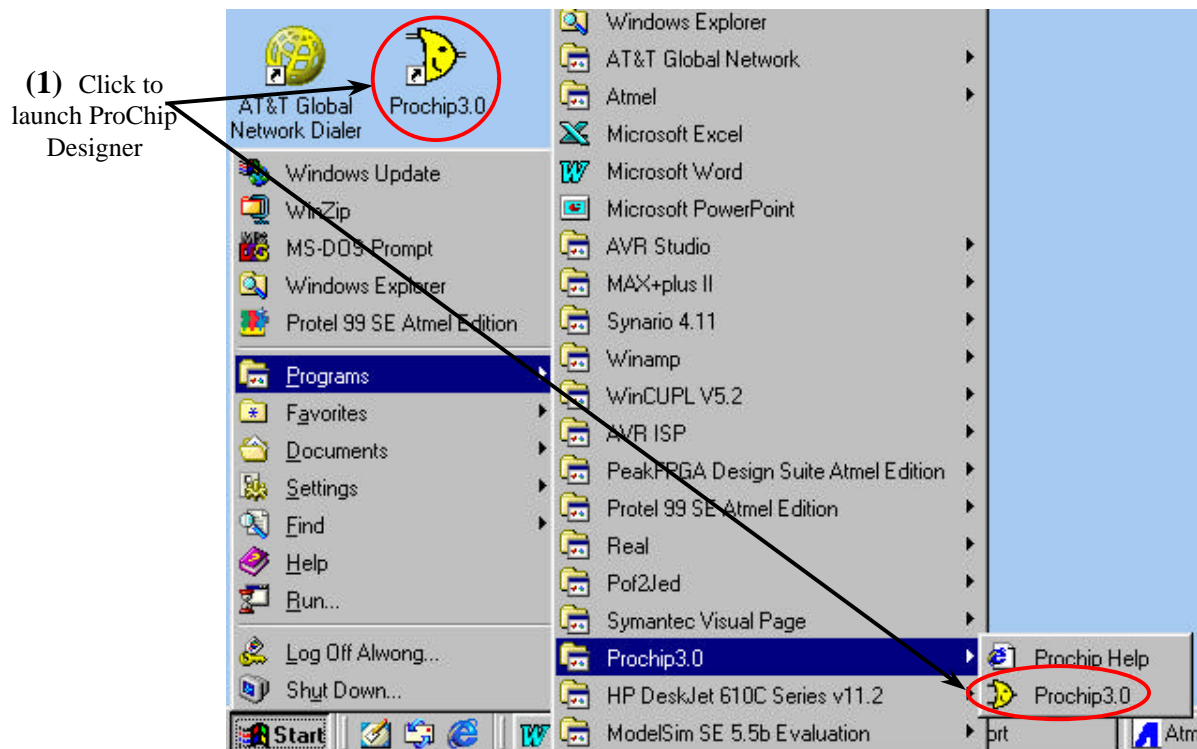
**Step IV: Use ATMEL Fitter to Fit Synthesized Design File**

**Step V: In-System Programming & Design Verification on CPLD Demo Board**

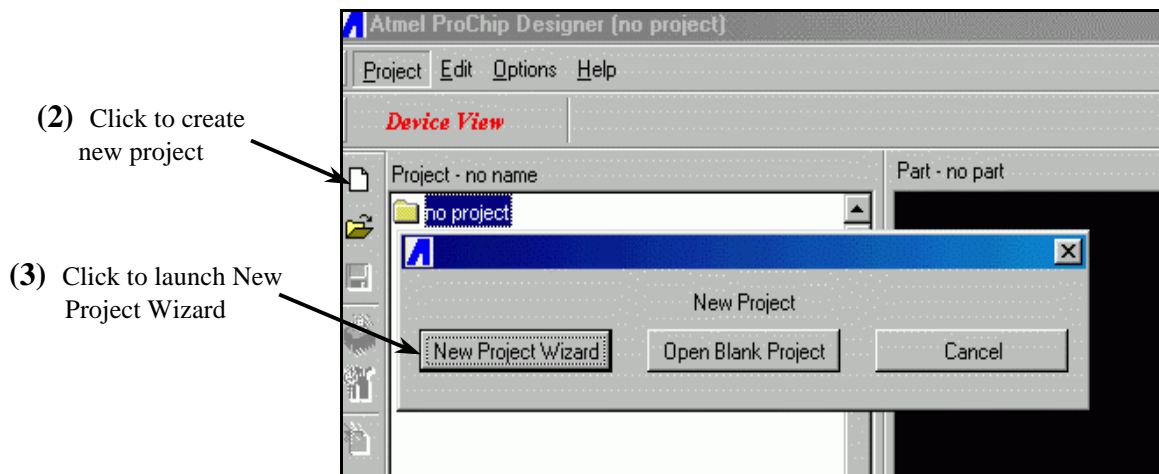
### Step I: Create a Project using the “New Project Wizard”

Before the design process can be started, a *Project File* must be created. ProChip Designer's **New Project Wizard** provides a very easy way to create a Project File.

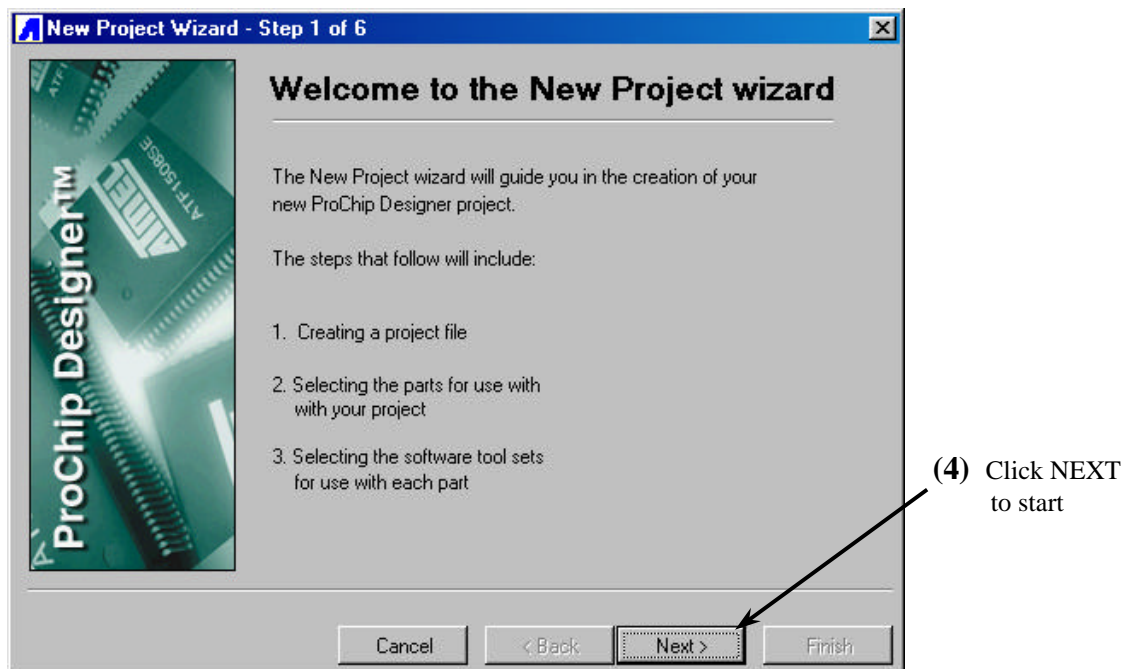
1. Click on the **START....PROGRAMS....PROCHIP** Icon to launch *ProChip Designer*.  
Or double-click on the **PROCHIP** icon on the desktop.



- Click on **PROJECT.....NEW** to launch the *New Project Wizard*.  
Or double-click on the **NEW PROJECT** shortcut button.
- Click on the **NEW PROJECT WIZARD** button to launch the *New Project Wizard*. This will launch the *New Project Wizard* to guide the users through the project creation process.

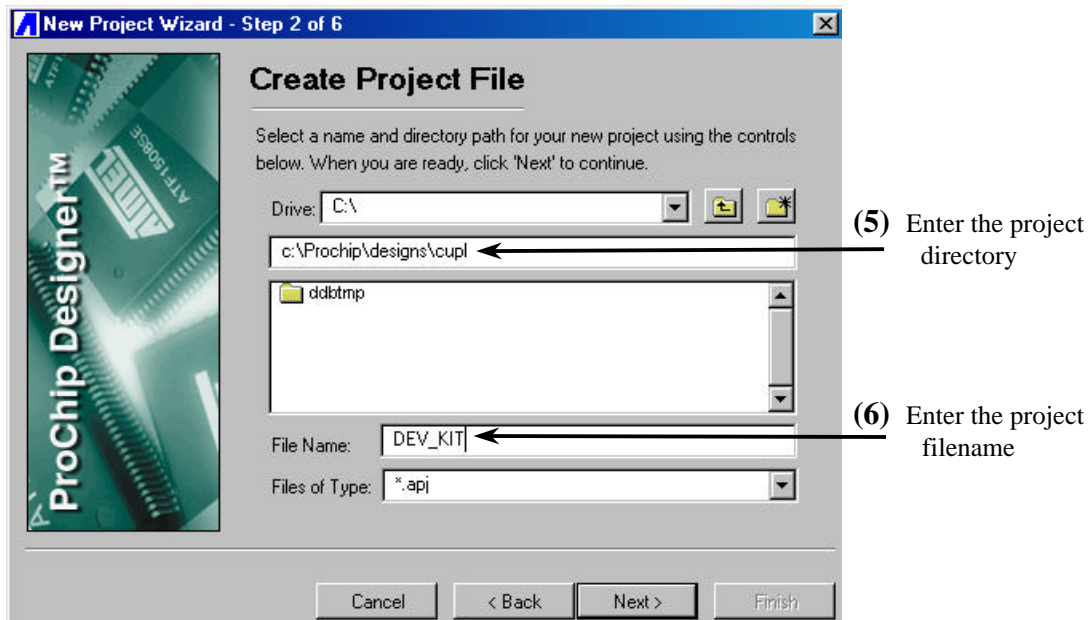


- Click on the **NEXT** button to start the project file creation process.

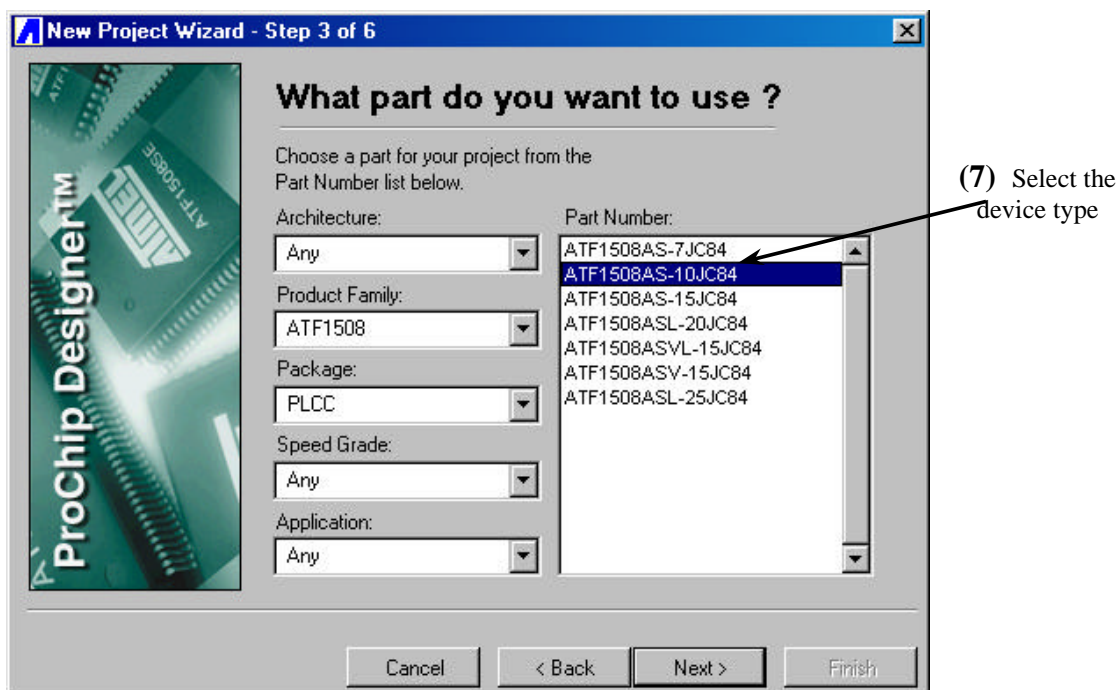


5. Use **C:\PROCHIP\DESIGNS\CUPL** as the directory of the project.
6. Enter **DEV\_KIT.APJ** as the project filename. The extension of a project file must be **.APJ**.

The name and directory of the design project is specified in this window. All design, simulation, and other project files must be placed in this project directory.

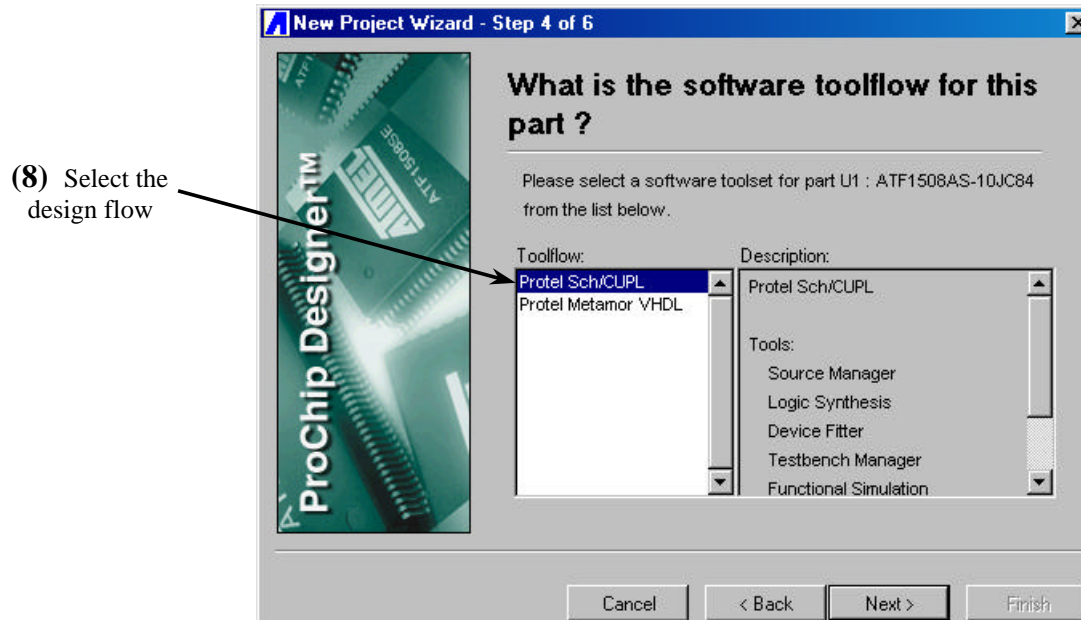


7. Choose a device for the project. **[ATF1508AS-10JC84]**. Also review the *Filters* that allow for selection of a specific *Speed Grade* or *Package Type*.



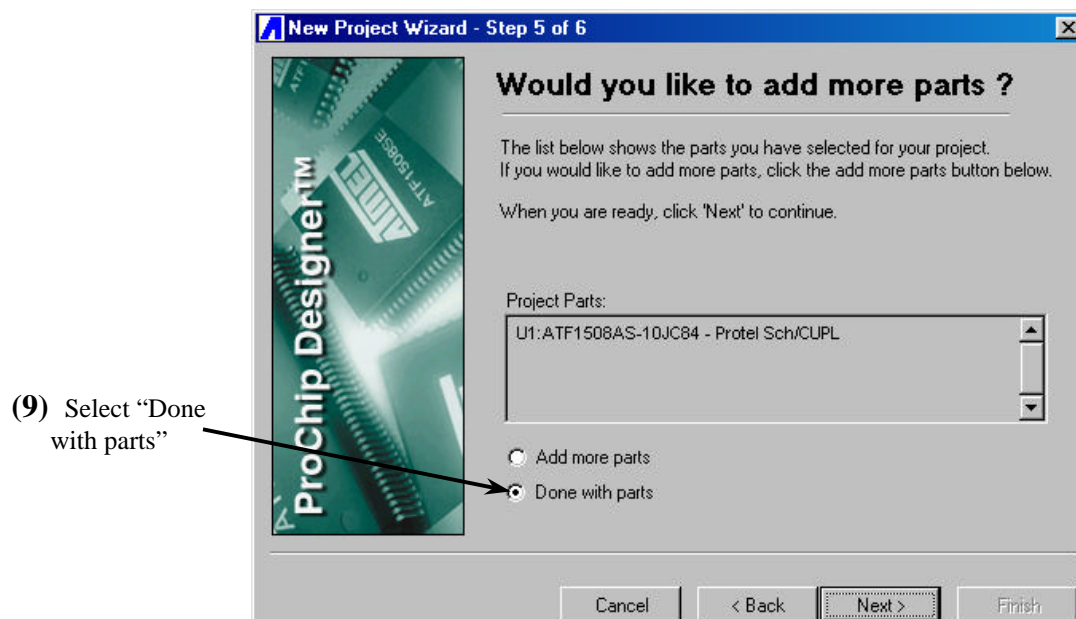
8. Select **PROTEL SCH/CUPL** as the software tool for this design flow.

**Protel Metamor VHDL and Protel Sch/CUPL** are the two possible design flows supported by ProChip Designer. The Protel Metamor VHDL design flow supports VHDL synthesis, VHDL functional simulation and VHDL timing simulation. The Protel Sch/CUPL supports Protel Schematic and CUPL design entries and function simulation.



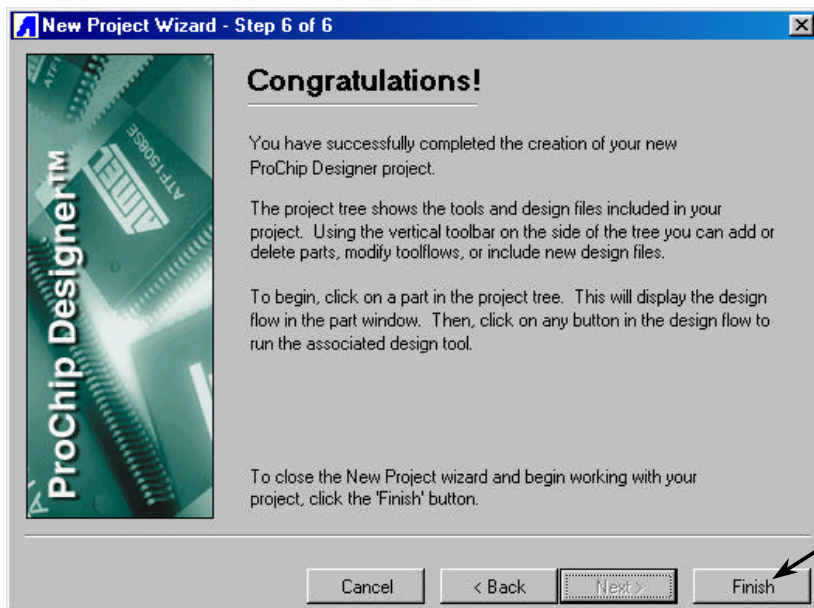
9. Select **DONE WITH PARTS** so that there will be only one device in this project.

On the other hand, users can select **ADD MORE PARTS** to include more parts to the current Project Directory.



10. Click the **FINISH** button to finish the *New Project Wizard* and the project creation process.

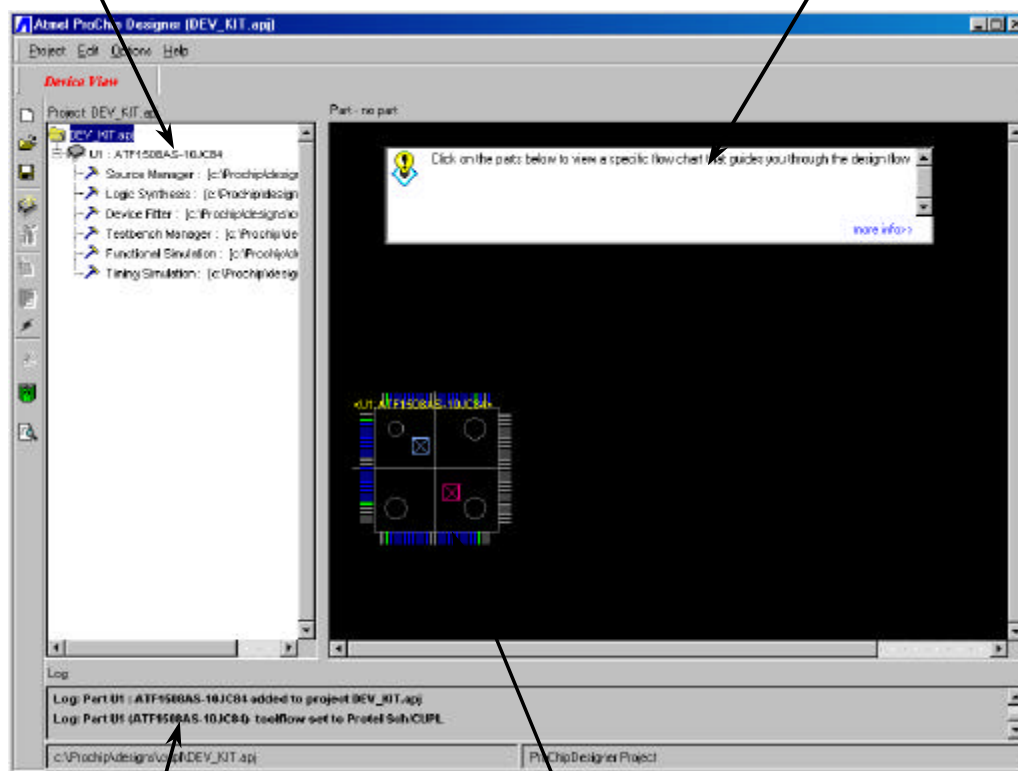
This closes the *New Project Wizard* and opens the *ProChip Designer* window. The *Sources* in the project are shown in the Left window.



(10) Click FINISH to end New Project Wizard

Project Sources Window

Information Dialog Box



Message Window

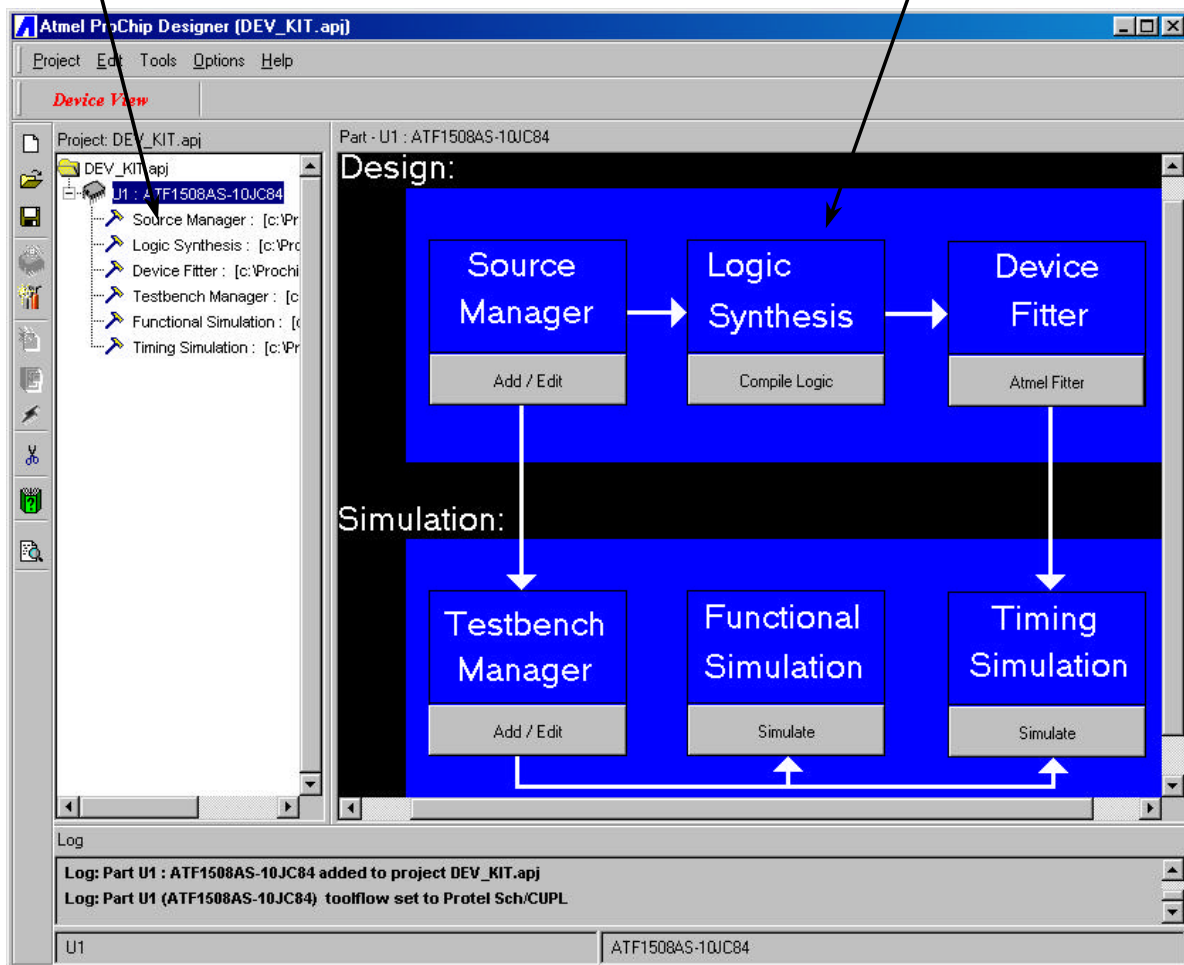
Project Device Icon



11. Click on the **Device Icon [ATF1508AS-10JC84]** to view the *Design Flow* window.

Design Processes

Design Flow Window



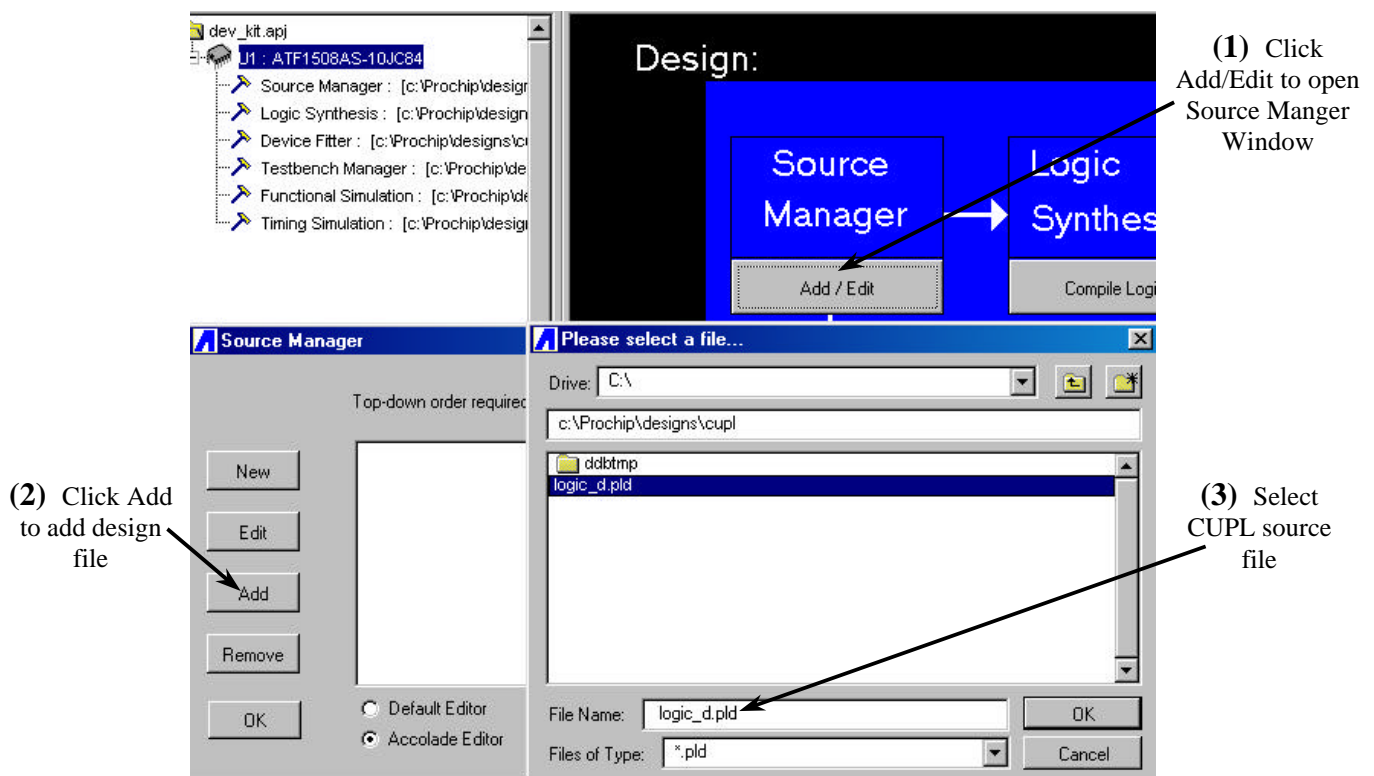
This completes Step I of this Tutorial.

---

## Step II: Add a CUPL Design File

Once the Project File is created, the next step is to add the design source file(s) into your project. For this tutorial, we will be adding a single CUPL design file into the project.

1. Click on the **ADD/EDIT** button from *Source Manager* to open the *Source Manager Window*. You can view the *Source Manager Help File* by clicking on the **Help** button within the *Source Manager Window* to view the description for the different processes.
2. In the *Source Manager Window*, click on the **ADD** button to add a CUPL design file to the project.
3. In the *File Manager Window*, select **LOGIC\_D.PLD** from the *c:\Prochip\designs\cupl* directory as the source design file for this project.



This “LOGIC\_D.PLD” is a CUPL design that uses the eight 8-segment LED displays and the built-in oscillator on the Atmel CPLD Demo Board to first display the word “logic” and then switches to the word “doubling”. This CUPL design can be compiled using either the ProChip Designer™ or the Atmel-WinCUPL® software.

The first section of the LOGIC\_D.PLD as shown below pre-defines which segments of the LED should be asserted in order to display the desired letter or number. For example, to display the letter “C”, segments A, D, E, and F need to be set to low and the remaining segments need to be set to high.

```

$define Font0 'b'1000000 /* = ( _f_e_d_c_b_a ); 0 */
$define Font1 'b'1111001 /* = ( _c_b ); 1 */
$define Font2 'b'0100100 /* = ( _g _e_d _b_a ); 2 */
$define Font3 'b'0110000 /* = ( _g _d_c_b_a ); 3 */
$define Font4 'b'0011001 /* = ( _g_f _c_b ); 4 */
$define Font5 'b'0010010 /* = ( _g_f _d_c _a ); 5 */
$define Font6 'b'0000010 /* = ( _g_f_e_d_c _a ); 6 */
$define Font7 'b'1111000 /* = ( _c_b_a ); 7 */
$define Font8 'b'0000000 /* = ( _g_f_e_d_c_b_a ); 8 */
$define Font9 'b'0011000 /* = ( _g_f _c_b_a ); 9 */

$define FontA 'b'0001000 /* = ( _g_f_e _c_b_a ); A */
$define FontC 'b'1000110 /* = ( _f_e_d _a ); C */

```

The next section of this PLD design as shown below shows you how to declare and assign pin numbers in the CUPL language to the input, output, and buried signals. The input and output pin assignments are assigned according to the connections between the CPLD and the eight 8-segment LED's as shown in the CPLD Demo Board schematic.

```

/* Inputs */
pin 1 = Gclr; /* Global Clear input */
pin 83 = Mclk; /* Global Clock input */

/* Outputs */
/* DSP1 */
pin 49 = LED1A; /* LED1 segment A */
pin 46 = LED1B; /* LED1 segment B */
pin 48 = LED1C; /* LED1 segment C */
pin 50 = LED1D; /* LED1 segment D */
pin 52 = LED1E; /* LED1 segment E */
pin 51 = LED1F; /* LED1 segment F */
pin 54 = LED1G; /* LED1 segment G */

```

Next the pinnode numbers are assigned to the buried signals as shown below. The feedback and/or the foldback paths available in each macrocell implement these buried signals. For the listing of the pinnode numbers, please refer to the “ATF15xx Device Help” section of the ProChip Designer Help File.

```

/* Nodes for the buried 22-bit Counter */
pinnode 603 = c0;
pinnode 605 = c1;
pinnode 606 = c2;
pinnode 608 = c3;

```

After assigning the input, output and buried signals, the related signals (e.g. the LED segments and buried counter) are grouped together as shown below to make the design source code more readable and easier to manage. In CUPL, the “Field” declaration can be used to group a specific set of signals.



```

Field DSP1 = [LED1G,LED1F,LED1E,LED1D,LED1C,LED1B,LED1A];
Field DSP2 = [LED2G,LED2F,LED2E,LED2D,LED2C,LED2B,LED2A];
Field DSP3 = [LED3G,LED3F,LED3E,LED3D,LED3C,LED3B,LED3A];
Field DSP4 = [LED4G,LED4F,LED4E,LED4D,LED4C,LED4B,LED4A];
Field DSP5 = [LED5G,LED5F,LED5E,LED5D,LED5C,LED5B,LED5A];
Field DSP6 = [LED6G,LED6F,LED6E,LED6D,LED6C,LED6B,LED6A];
Field DSP7 = [LED7G,LED7F,LED7E,LED7D,LED7C,LED7B,LED7A];
Field DSP8 = [LED8G,LED8F,LED8E,LED8D,LED8C,LED8B,LED8A];

Field Count = [c21..c0] ;

```

Next a 22-bit buried counter using T-type Flip-flops is implemented as shown below to divide the 2.0MHz clock into a much slower frequency signal that can be used to display the text messages. The last bit of this counter (c21) will be a periodic square wave signal oscillating at 0.477Hz ( $2\text{MHz} \div 2^{22} = 2 \times 10^6 \div 4194304 = 0.477\text{Hz}$ ).

```

/* Frequency Divider */
c0.d = !c0;
c1.t = c0;
c2.t = c1 & c0;
c3.t = c2 & c1 & c0;
:
:
c20.t = c19 & c18 & c17 & c16 & c15 & c14 & c13 & c12 & c11 & c10 & c9
& c8 & c7 & c6 & c5 & c4 & c3 & c2 & c1 & c0;
c21.t = c20 & c19 & c18 & c17 & c16 & c15 & c14 & c13 & c12 & c11 &
c10 & c9 & c8 & c7 & c6 & c5 & c4 & c3 & c2 & c1 & c0;

Count.ck = Mclk;
Count.ar = !Gclr;

```

Finally, the last section of the PLD design is to assign the eight 8-segment LED displays to the appropriate letters or numbers as shown below to display the words “logic” and “doubling”. The word “logic” is displayed when c21 is low and then “doubling” is displayed when c21 is high. You can easily change the letters/numbers to be displayed by changing this section of the code to the appropriate pre-defined letters/numbers.

```

DSP8 = Fontl & !c21 # Fontd & c21;
DSP7 = Fonto & !c21 # Fonto & c21;
DSP6 = Fontg & !c21 # Fontu & c21;
DSP5 = Fonti & !c21 # Fontb & c21;
DSP4 = Fontc & !c21 # Fontl & c21;
DSP3 = FontBK & !c21 # Fonti & c21;
DSP2 = FontBK & !c21 # Fontn & c21;
DSP1 = FontBK & !c21 # Fontg & c21;

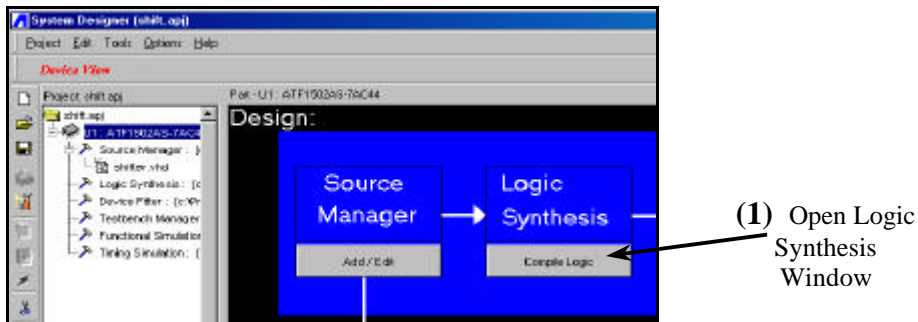
```

You have now completed Step II of this Tutorial.

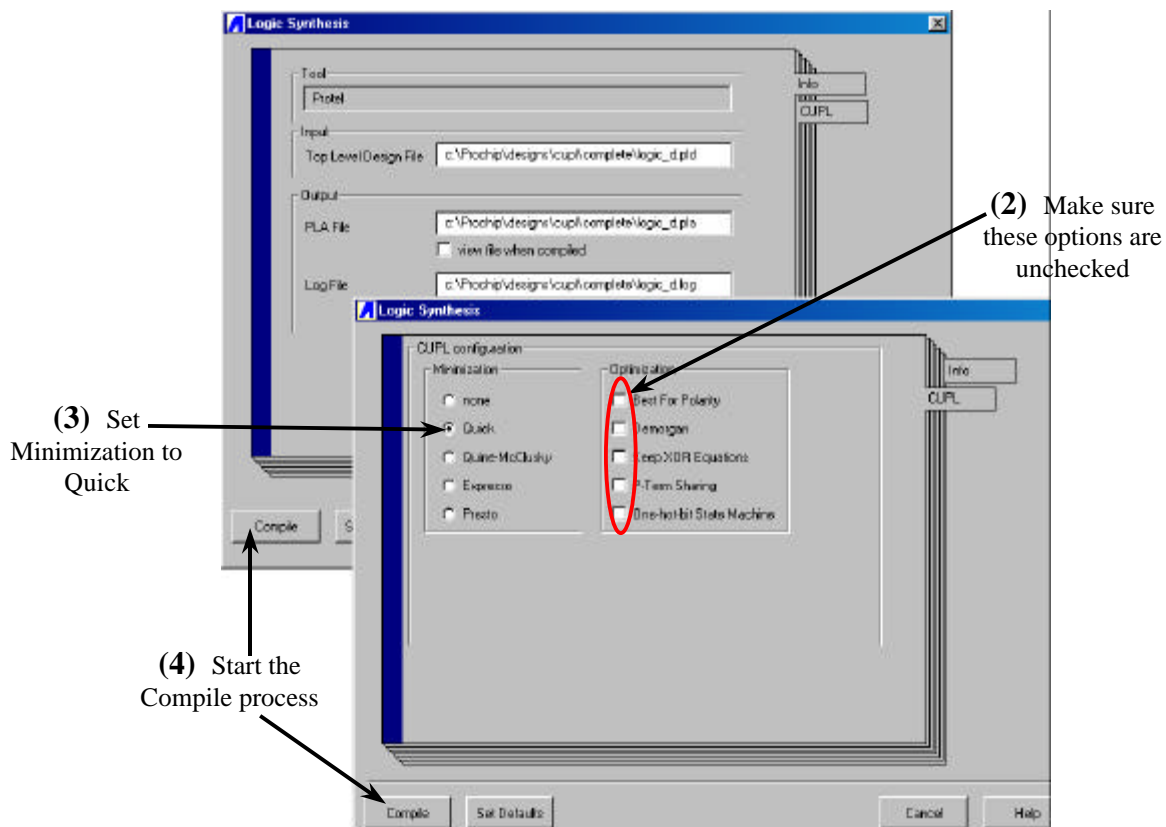
## Step III: Compile the CUPL Design

In this step, you will compile your CUPL design through Protel Design Explorer 99 SE into a set of optimized/minimized logic equations.

1. Click on the **COMPILE LOGIC** button in the *Design Flow Window* to open the *Logic Synthesis Window*.



2. Make sure all of the options in the *Optimization* section are unchecked.
3. Make sure the *Minimization* setting is set to **QUICK**.
4. Click on the **COMPILE** button to start the CUPL compile process.



You can click on the **SET DEFAULTS** button and it will automatically specify the Synthesis tool in the Tool Text box.

If you click on the CUPL Tab, it shows the various Synthesis options. You may refer to the HELP file for further description.

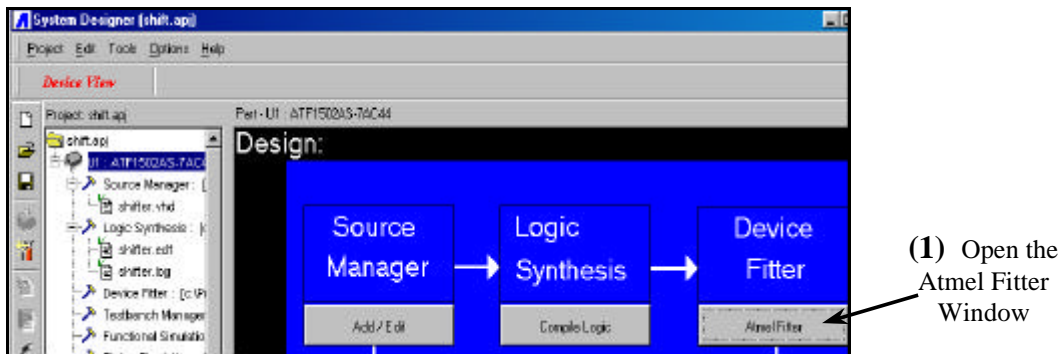
This completes Step III of this Tutorial.

---

## Step IV: Use ATMEL Fitter to Fit the Synthesized Design File

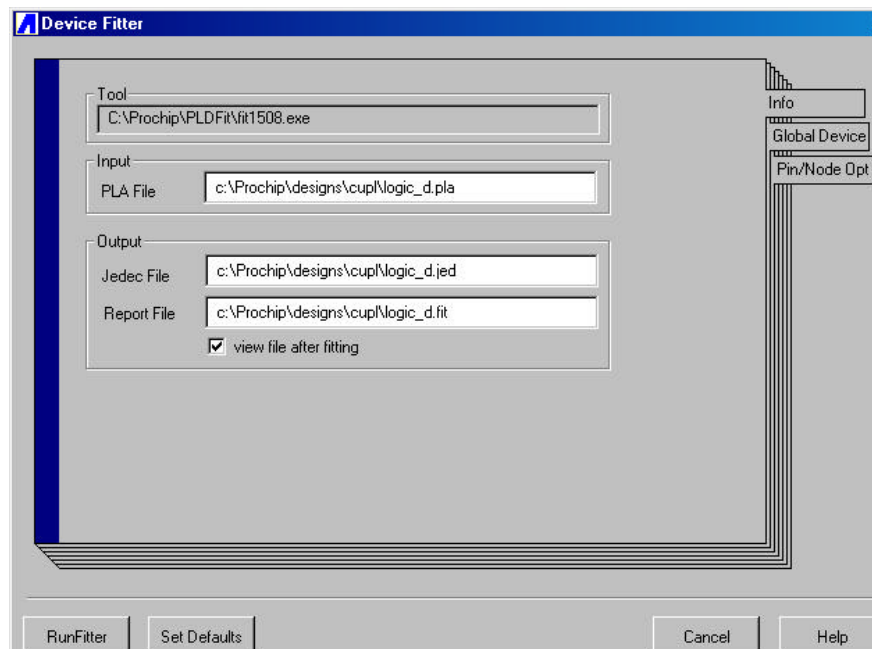
In Step III, we completed the Logic Synthesis portion of the Design Flow. On successful compilation, the CUPL compiler tool produces a PLA output file (with extension .pla). A PLA file contains the netlist of the optimized and minimized logic equations. We now need to map this netlist into a specific Atmel PLD architecture using the **ATMEL FITTER**.

1. You can now proceed to the *Device Fitter* portion of the Design Flow by clicking on the **ATMEL FITTER** button.



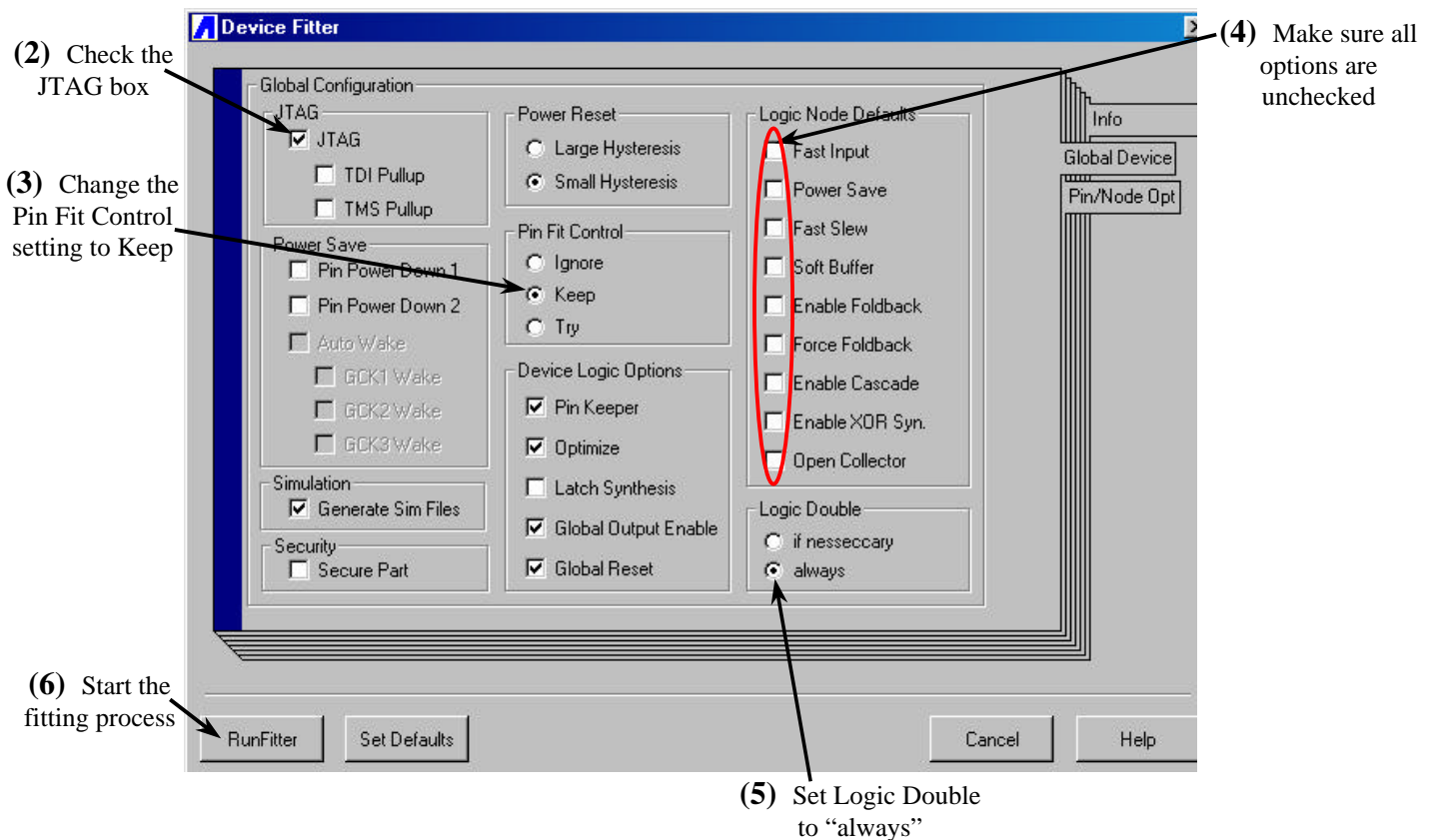
You can either use the Default options or specify Fitter properties. You must select a PLA file or the Fitter will not run. In this example, since our target device is an ATF1508AS, and we will select the fit1508.exe device fitter.

The fitter creates the important JEDEC and FIT REPORT output files. They contain the data for programming the Device (using In-System Programming or on a third party device programmer) and the pin assignments required for board layout respectively.



Please review the *Global Device Parameters* and *Pin/Node Options* as well. The *Help Files* also show the Device Pin\_Node lists for each of the ATMEL CPLDs.

2. Make sure the **JTAG** box is checked. This enables the JTAG port for ISP programming.
3. Make sure the **PIN FIT CONTROL** setting is set to **KEEP**. This will ensure that the pin assignments in the PLD file will be kept during the Place-and-Route process.
4. Make sure all of the options in the **LOGIC NODE DEFAULTS** section are unchecked.
5. Change the **Logic Double** control to *always*. This enables the fitter to use ATF15xx Family devices Logic Doubling features beginning in the first fitter pass.
6. When all the fitter options are set, click on the **RUN FITTER** button to fit the design.



The Fitter Report File is shown below.

```
Atmel ATF1508AS Fitter Version 1.8.3.9 ,running Thu Jul 05 19:30:31 2001

fit1508 -i C:\FITTER\TEST\V1839\LOGIC_D\LOGIC_D.tt2 -CUPL -dev P1508C84 -JTAG ON

***** Initial fitting strategy and property *****
Pla_in_file = LOGIC_D.tt2
Pla_out_file = LOGIC_D.tt3
Jedec_file = LOGIC_D.jed
Vector_file = LOGIC_D.tmv
verilog_file = LOGIC_D.vt
:
:
:
** Resource Usage **

    pin_num pin_name output_type feedback foldback cascade_out TotPT output_slew
MC1      0    --      --      --      --      --      0    slow
MC2      0    --      --      --      --      --      0    slow
MC3     12  LED6B    reg      c0      --      --      2    slow
MC4      0    --      --      --      --      --      0    slow
MC5     11  LED5G    reg      c1      --      --      3    slow
MC6     10  LED5E    reg      c2      --      --      3    slow
MC7      0    --      --      --      --      --      0    slow
MC8      9  LED5F    reg      c3      --      --      3    slow
MC9      0    --      --      --      --      --      0    slow
MC10     0    --      --      --      --      --      0    slow
:
:
```

ATF15xx Family devices Logic Doubling features provide extra IO connectivity and logic reusability. For example, every pin has the option of individual output enable, and every macrocell has the option of feedback of either the registered or combinatorial output independently of which signal is routed to the pin driver.

In the LOGIC\_D example given here, Macrocells 3, 5, 6 etc. route their combinatorial outputs, LED6B, LED5G, LED5E, etc. to the pins while also routing their registered outputs, c0, c1, c2 etc. to the feedback outputs, thus doubling the utilization of the macrocell logic.

For more examples of design techniques that utilize the Logic Doubling features of the ATF15xx Family, refer to Atmel's Logic Doubling White Paper and Reference Designs, available on the Atmel website at <http://www.atmel.com/atmel/products/prod147.htm#refdsdns.zip>. These examples show how to apply Logic Doubling techniques to new product designs, and obtain the benefits of more features in a smaller, and possibly less expensive chip, or spare logic resources for future revisions and reduced risk of PCB re-spin.

This completes Step IV of this Tutorial.



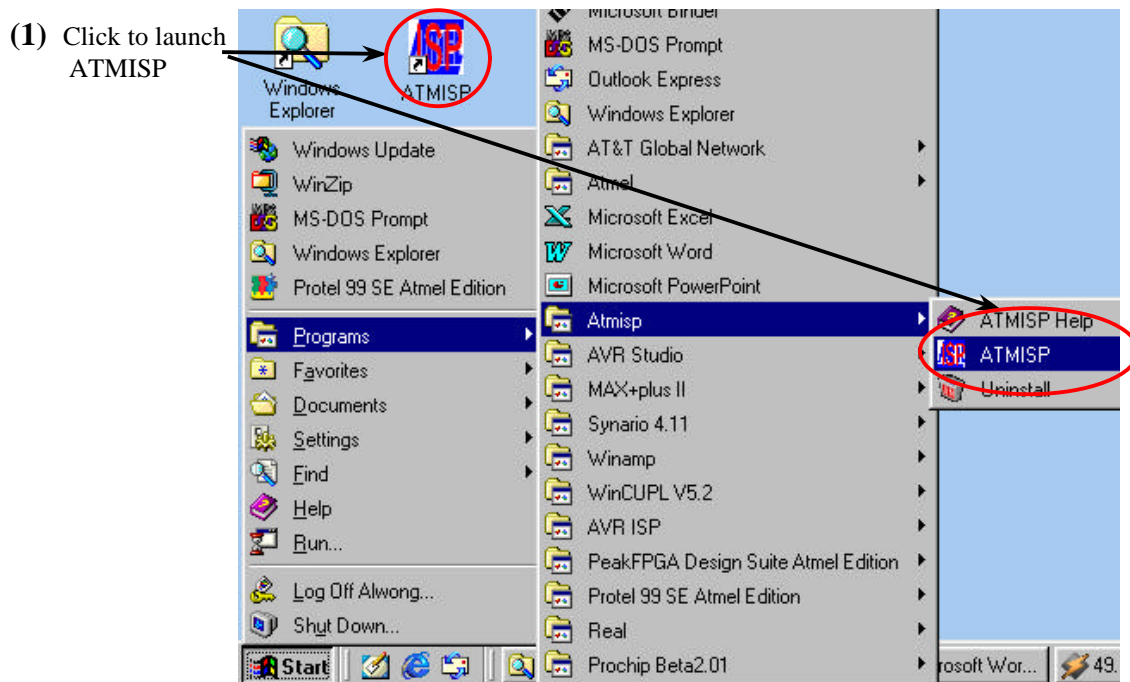


## Step V: In-System Programming and Design Verification

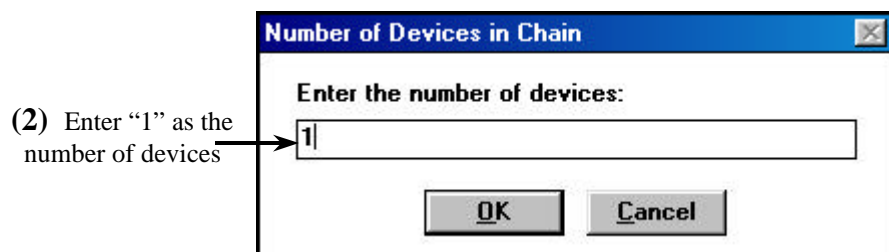
In this step of the tutorial, you will program an ATF1508AS 84-pin PLCC device on the Atmel CPLD Demo Board through ISP and then verify the design by observing the text messages displayed on the eight 8-segment LED displays on the Development Kit.

You will need to initialize the ATMISP software in order to program the ATF1508AS 84-pin PLCC on the CPLD Demo Board.

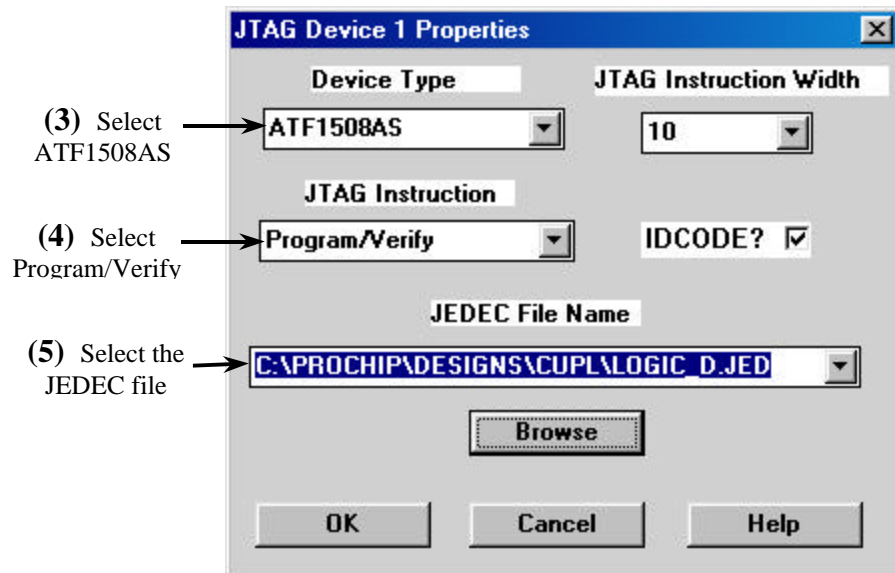
1. Click on the **START....PROGRAMS....ATMISP Icon** to launch the *Atmel-ISP Software*  
Or double-click on the **ATMISP** icon on the desktop.



2. Enter “1” as the number of devices in your JTAG device chain and then click on the **OK** button to proceed to the next step.

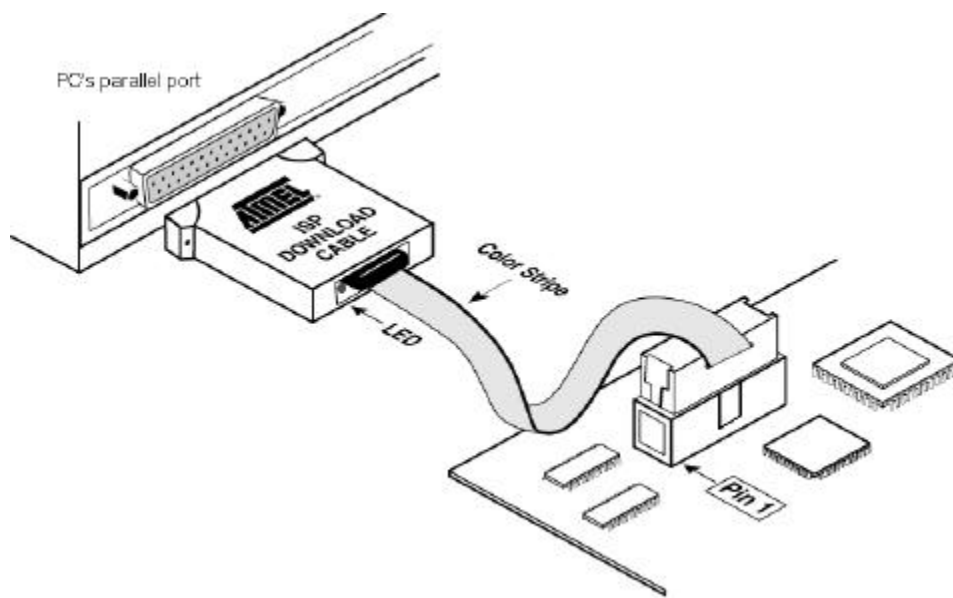


3. Select **ATF1508AS** as the target device type.
4. Select **PROGRAM/VERIFY** as the JTAG Instruction type.
5. Select “**C:\PROCHIP\DESIGNS\CUPL\LOGIC\_D.JED**” as the JEDEC file to be programmed into the ATF1508AS and then click on the **OK** button to proceed to the next step.

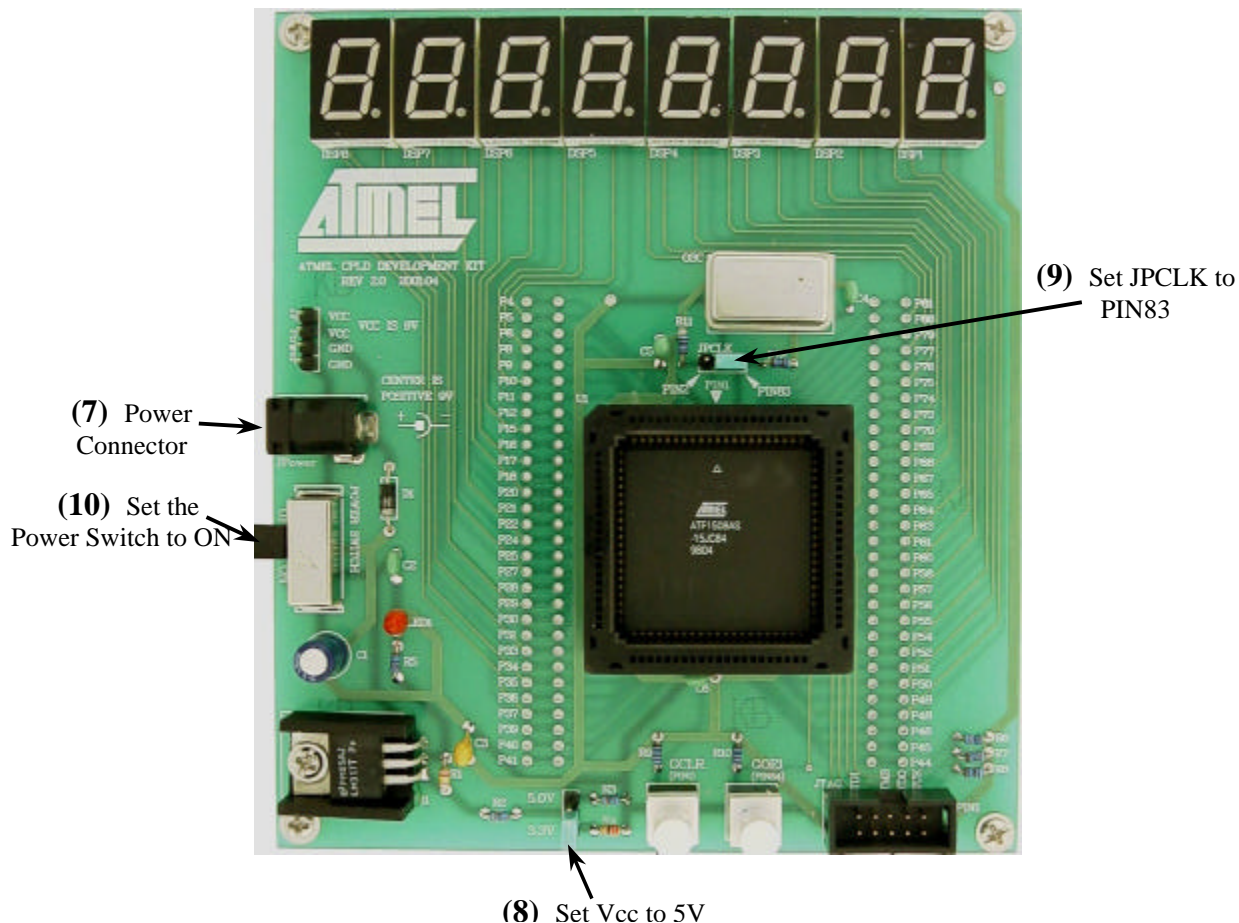


The next step requires you to setup the Atmel CPLD Demo Board to program the ATF1508AS through ISP.

6. Connect the 25-DB side of the Atmel-ISP Cable to the PC's parallel port and the 10-pin header side to the Atmel CPLD Demo Board as shown below.

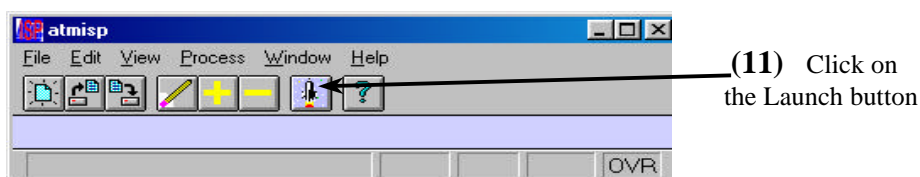


7. Connect a 9V AC/DC power adapter to the power connector (JPower) of the Atmel CPLD Demo Board.
8. Set the 5V/3.3V jumper to **5V** to set the system Vcc to 5V.
9. Set the JPCLK jumper to **PIN83** so that the output of the crystal oscillator will go to Pin 83 of the ATF1508AS.
10. Switch the Power Switch to the **ON** position.



Now both your software and hardware are setup for ISP programming, and you can execute the PROGRAM/VERIFY instruction to program the ATF1508AS on the Atmel CPLD Demo Board.

11. Click on the LAUNCH button in the ATMISP main window to execute the JTAG instruction to program the ATF1508AS on the Development Kit.



After the successfully programming the ATF1508AS with the LOGIC\_D.JED file, the eight 8-segment LED's should first display the word "logic" and then the word "doubling".

If these two text messages are correctly displayed on the Development Kit, then you have successfully completed this tutorial.

Congratulations!

---