Altera Development Board **Supplement Version 2.0**

The Altera Development Board contains a Programmable Logic Device (PLD) which is a chip that can be programmed to perform a wide variety of logical operations (AND, OR, NOT, NAND, etc) in any combination. The PLD provided on our development board (EPM7128S) can be programmed to represent up to 2,500 logic gates.

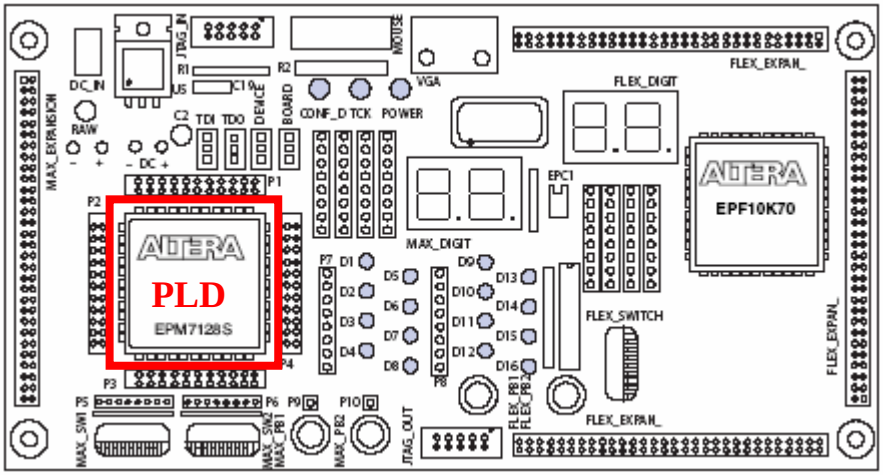


Figure 1. Altera Development Board

Quartus II Software

The Quartus II software provides the means in which we can program the PLD with our circuit design. The software facilitates a process in which we create our design, verify our design is correct, and then implement our design with the PLD. The basic design flow is shown in Figure 2. This software allows you to build digital logic circuits with pictures. Next the program will check to see if there are any errors (we call this compiling). When you compile your design, the program determines how to configure the PLD so that it functions just as you've designed. Once your circuit has compiled with no errors, you can then simulate it and the program will display an output for your circuit. This will allow you to see if the circuit is doing what you thought it would. A by-product of compiling our design, is that the program will select certain pins on the PLD for the inputs/outputs. However, you sometimes know better where you would like to connect these than the program so you have the option of assigning the pins ourselves. Finally, you program the PLD by sending a stream of bits from your program to the PLD which contains the information on how to configure the PLD so that it functions just as you've designed. To check our design, we connect the chip inputs to switches and toggle those on and off for "1s" and "0s" and connect the chip outputs to LEDs which when on or off represent a "1" or "0" respectively.

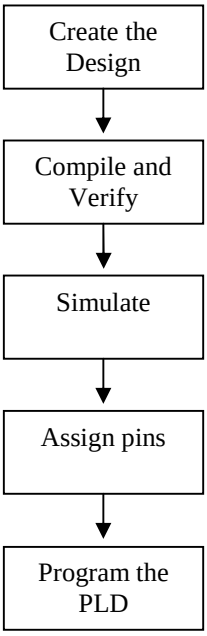


Figure 2. Design Flow

I. Getting Started

1. Download Quartus 6.1 Web Edition from: (you want 6.1 instead of 7.1 so it will match the lab) https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we-v61.jsp
   a. Download and install (but do not run)
2. Download and License Setup (This step may not be necessary.)
   a) https://www.altera.com/support/licensing/free_software/lic-q2web.jsp
   b) Create account and sign in. (you'll need your MAC address i.e. physical address)
   c) To find NIC, click start click run type "cmd" type "ipconfig /all" physical address is NIC.
   d) Enter MAC and then download license and save it in: C:\altera\61\quartus
   e) Start program …. it may ask you for information for the license file, if so, point it to the file you just previously saved.

Setup a new project in Altera Quartus II.

a. Select **File → New Project Wizard**
b. Enter the directory name and project name as indicated below. Do not include spaces or symbols in the file name. *Every name can have a maximum of 8 characters with no symbols other than underscores allowed.*

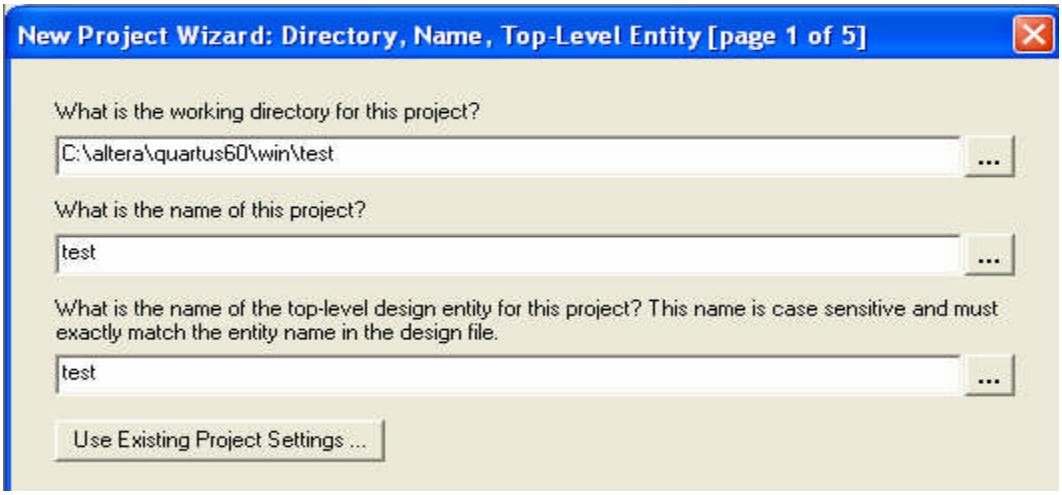


Figure 1

c. Click **Next**. If the directory does not already exist, a dialog will ask to create it. Click **Yes**.
d. Click **Next** 2 more times.
*e. Select Family **MAX7000S,** and click **Next**.*
f. Select *Speed Grade* **7**, *Pin Count* **84**, **EPM7128SLC84-7** in the *Available Devices* list box, and click **Finish**. (If **EPM7128SLC84-7** isn't available, select **EPM7128SLC84-6.)**

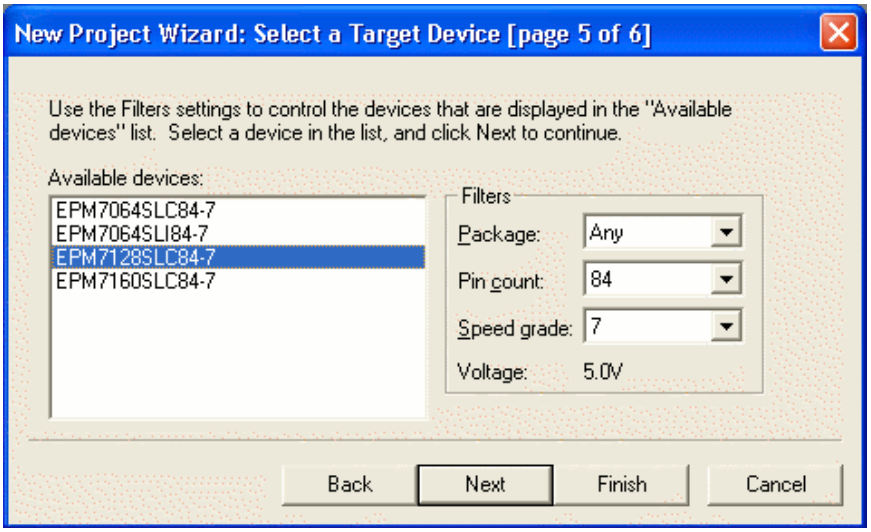

New Project Wizard: Select a Target Device [page 5 of 6]
Use the Filters settings to control the devices that are displayed in the "Available devices" list. Select a device in the list, and click Next to continue.
Available devices:
EPM7064SLC84-7
EPM7064SLI84-7
EPM7128SLC84-7
EPM7160SLC84-7
Filters
Package:
Any
Pin count:
84
Speed grade:
7
Voltage:
5.0V
Back
Next
Finish
Cancel

Figure 2

II. Drawing a block diagram.

3. Adding your new Device Schematic to the Project
   a. Select **File → New**
   b. Select **Block Diagram/Schematic File**
   c. Click the **OK** button.

4. Draw the schematic.
   a. Double click the area where you want a new item.
   b. Select the item you want from the Libraries List Box. To expand or contract the list click on the plus or minus sign respectively. Note that most items you need will be in the *primatives/logic* or *primatives/pin* folder. Alternatively, you may type the name of the item you want in the **Name Box** from Table 1.

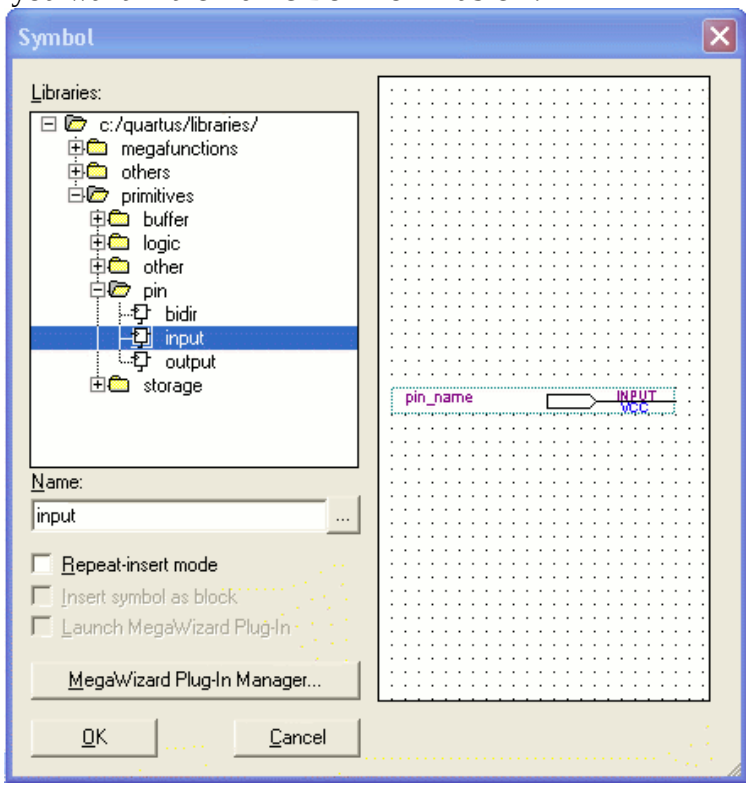
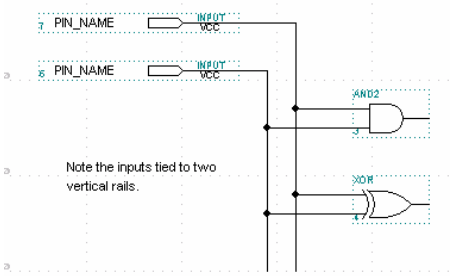


Figure 3

Device/Gate	Text Entry
Input	`input`
Output	`output`
2 Input AND	`and2`
3 Input AND	`and3`
2 Input OR	`or2`
Inverter	`not`
XOR	`xor`
2 Input NOR	`nor2`
JK FlipFlop	`jkff`
D FlipFlop	`dff`
T Flip Flop	`tff`
Freq Divider	`freqdiv`
Nor Latch	`norltch`
Ground (low)	`gnd`
Vcc (high)	`vcc`

Table 1

   c. Click **OK.**
   d. Place the object in the desired location on the schematic and click the **Left Button**.
   e. Give each input and output a unique name by editing the pin name. You can edit the pin name by **Single** or **Double Left Clicking** on top of the name.
   f. Connect the devices using the orthogonal node tool to implement the circuit. (Note: this is much easier to do if you take each input and tie it a vertical rail as shown below in Figure 4).
   g. Save the file using **File → Save As**. *The filename must be 8 characters max with no spaces.*



Don't forget to add an output block.

Figure 4

5. Save the project.
    a. Select **File → Save Project**.
6. Compile the Project
    a. Press **Ctrl+L** or Select **Processing → Start Compilation**
    b. If all went well, I dialog box will indicate compilation complete. Otherwise, the compiler report should indicate what the errors were.

III. Simulating the project with the waveform editor.

7. Simulate the project.
   a. Select **Processing → Simulator Tool**.
   b. Set the *Simulator Mode* to **Functional** as in Figure 5b.
   c. Click the **Open** Button.
   d. Select **Edit → Insert Node or Bus**.
   e. Click the **Node Finder** Button.
   f. Click on **Filter → Pins: all**.
   g. Click the **List** Button.
   h. Click the >> Button.
   i. Click **OK** Button.
   j. Click **OK** Button.
   k. Highlight all the Inputs per Figure 5a.

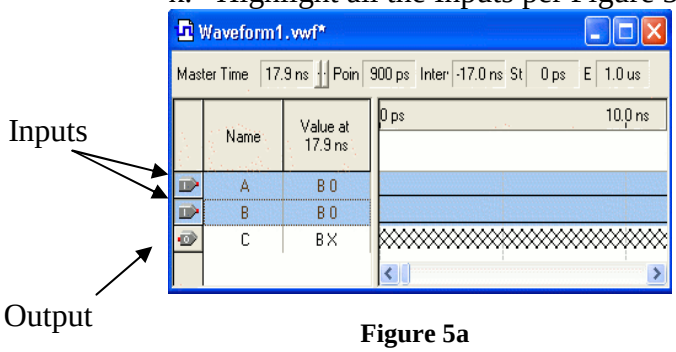


Figure 5a

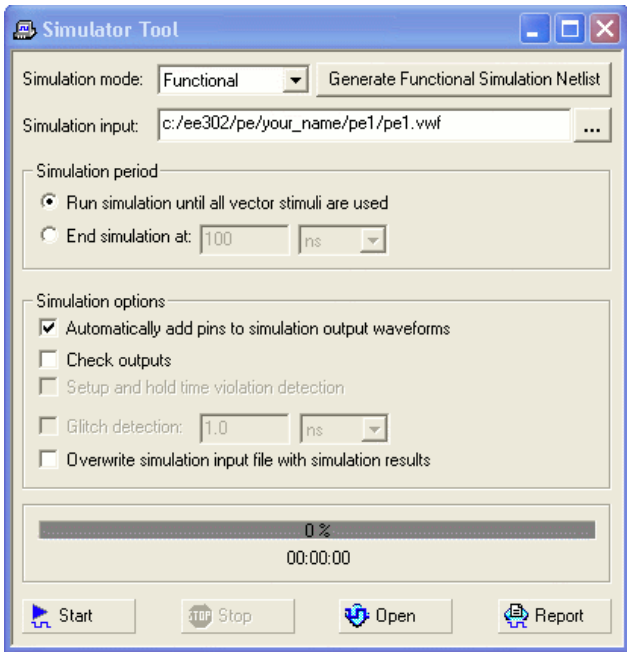


Figure 5b

   l. Select **Edit → Group**.
   m. Enter **Input** for the *Group Name*.

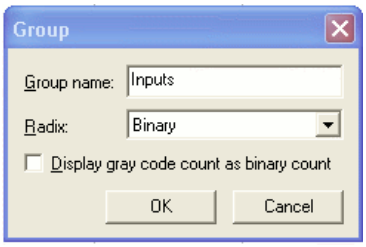


Figure 6

   n. Click **OK**.
   o. Select the **Inputs** on the Waveform window.
   p. Click the **Count** Button.
   q. Click **OK**.
   r. Select **File → Close**.
   s. Click the **Save** Button.
   t. *Click the **Generate Functional Simulation Netlist** on the Simulation Tool.*
   u. *Add the Simulation input: <name>.vwf (Use the browse feature)*
   v. Click the **Start** Button on the Simulator Tool.
   w. Click **OK**.
   x. Click the **Report** Button on the Simulator Tool. If there were no simulation errors, the Input and Output waveforms should be displayed.

IV. Pin Assignments

8. Assign Pins to Inputs and Outputs
   a. Select **Assignments → Assignment Editor**.

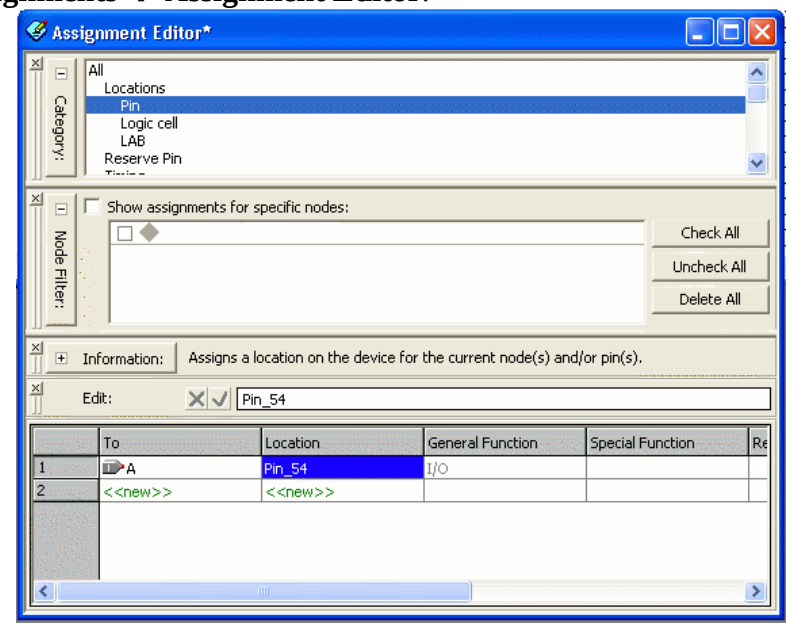


Figure 7

   b. Select the **Pin** *Category* in the Assignment Editor per Figure 7.

Important Note: Write down the pin numbers assigned in the next step!

   c. Perform the following 4 steps for each Input and Output.
      i. Double Click **<<new>>** in the *To* Column
      ii. Select the desired Input or Output
      iii. Double Click **<<new>>** in the *Location* Column
      iv. Select the desired pin number. *Choose a pin from pin row P1 or P2 (See Figure 9.)*
   d. When complete select **File → Close**.
   e. Click the **Yes** Button.

Note: For output to pins of the 7 Segment Display refer to Table 2.

Display Segment	Output pin for Digit 1	Output pin for Digit 2
a	58	69
b	60	70
c	61	73
d	63	74
e	64	76
f	65	75
g	67	77
Decimal	68	79

Table 2

I/O Pin Assignments	Pin #
Clock (Input Pin)	83
Not Available	12, 33, 54, 75

Table 3

V. Programing the Altera Board.

9. Check the JTAG jumpers on the Altera board.
    a. All four jumpers should be oriented on the top two pins of each column. See Figure 9.

10. Implement the design on the Altera board.
    a. Connect the power and parallel cables to the Altera board.
    b. Select **Tools → Programmer.**
    c. Select **Program/Configure** Check Box.
    d. Click the **Start** Button.
    *e. If the **Start** Button is not available, click the **Hardware Setup** Button. Select **Add Hardware**, Byte Blaster II, **Ok**, **Select Hardware** and **Close**.*
    f. Disconnect the parallel cable and the power cable (unplug from the wall, not the board).
    g. Wire the necessary wires from switches to the Altera pins and from the Altera pins to the output LEDs, 7-Segment Display, etc. Use the pin numbers you wrote down from step 7. Figure 8 shows the pin number for the Altera Chip.
    h. Wiring notes.
        i. All LED's are Active Low (low turns them on, high turns them off).
        ii. For the switches, open=high and closed=low.
        iii. The seven segment LED displays are hardwired to output pins and are Active Low.
    i. Reconnect the power cable and test your circuit.

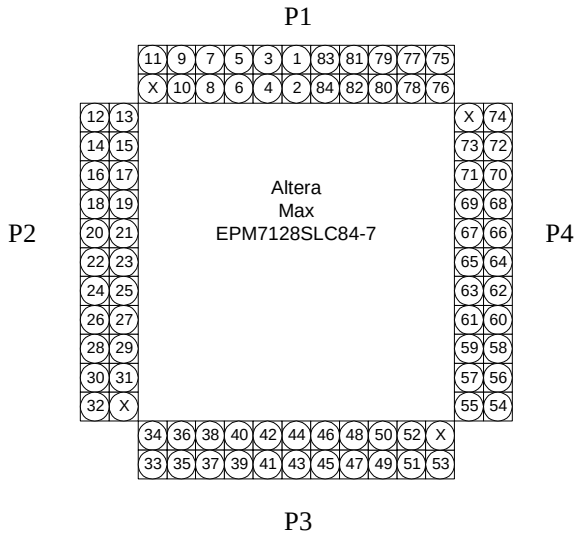


Figure 8

11. Creating a symbol block.
    a. Open the Schematic for the entity you want to create a symbol for.
    b. Select **File → Create/Update → Create Symbol Files for Current File**
    c. Click **OK**

12. Using a symbol block from another project.
    a. Using Windows Explorer, copy the files ***device_lastname.bsf*** and ***device_lastname.bdf*** from the old project directory to the new project directory.
    b. The device will be available in the *Symbol* dialog box under the *Project* folder.

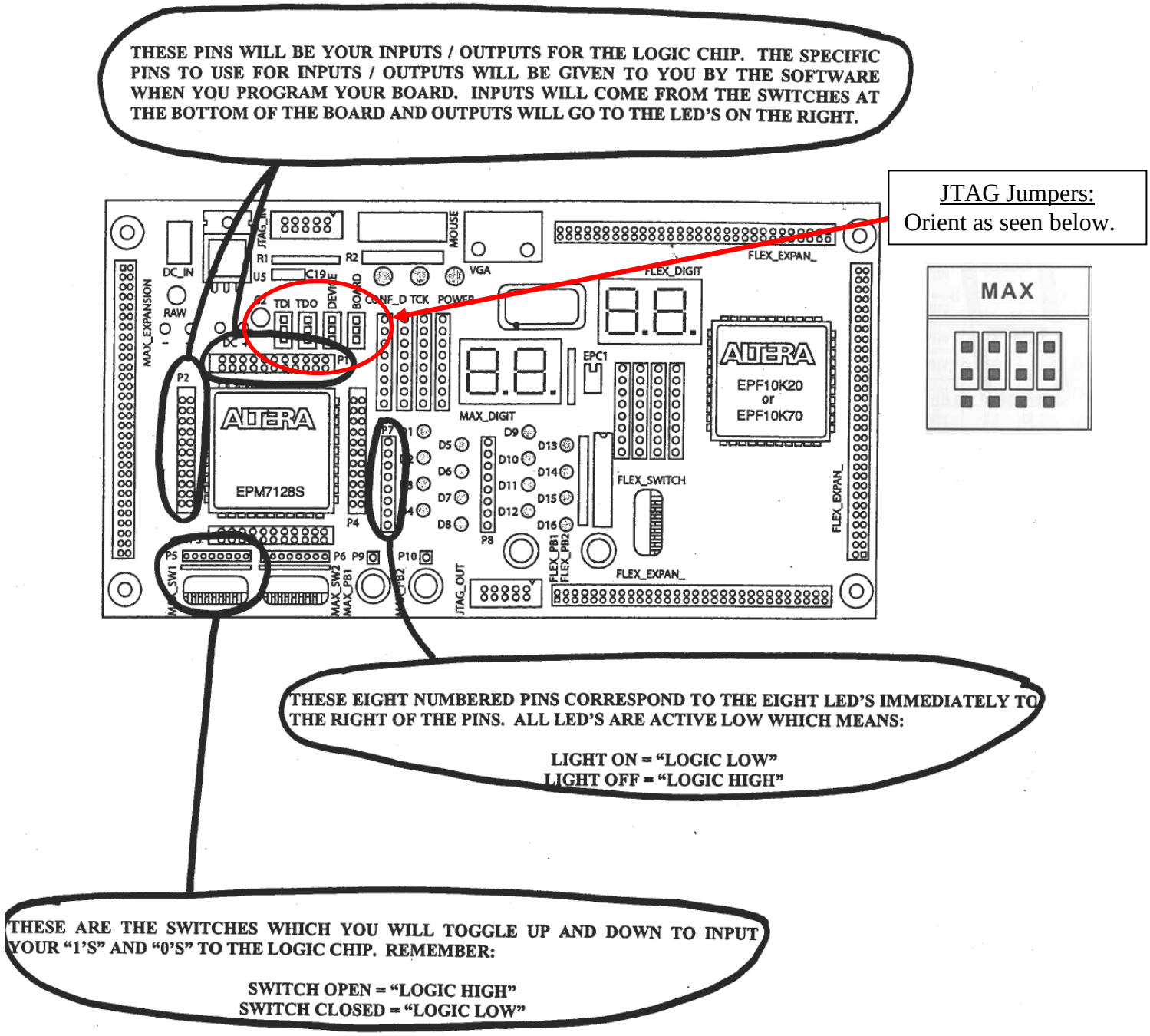

THESE PINS WILL BE YOUR INPUTS / OUTPUTS FOR THE LOGIC CHIP. THE SPECIFIC PINS TO USE FOR INPUTS / OUTPUTS WILL BE GIVEN TO YOU BY THE SOFTWARE WHEN YOU PROGRAM YOUR BOARD. INPUTS WILL COME FROM THE SWITCHES AT THE BOTTOM OF THE BOARD AND OUTPUTS WILL GO TO THE LED'S ON THE RIGHT.
JTAG Jumpers:
Orient as seen below.
MAX
THESE EIGHT NUMBERED PINS CORRESPOND TO THE EIGHT LED'S IMMEDIATELY TO THE RIGHT OF THE PINS. ALL LED'S ARE ACTIVE LOW WHICH MEANS:
LIGHT ON = "LOGIC LOW"
LIGHT OFF = "LOGIC HIGH"
THESE ARE THE SWITCHES WHICH YOU WILL TOGGLE UP AND DOWN TO INPUT YOUR "1'S" AND "0'S" TO THE LOGIC CHIP. REMEMBER:
SWITCH OPEN = "LOGIC HIGH"
SWITCH CLOSED = "LOGIC LOW"
EPM7128S
EPF10K20
or
EPF10K70
ALTERA
MAX_EXPANSION
FLEX_EXPAN_
FLEX_DIGIT
MAX_DIGIT
FLEX_SWITCH
JTAG_OUT
MOUSE
VGA
DC_IN
RAW
TDI TDO
CONF_D TCK
POWER
EPC1
P2
P4
P5
P6
P8
P9
P10
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
MAX_SW1
MAX_SW2
MAX_PB1
MAX_PB2
FLEX_PB1
FLEX_PB2

Figure 9

Appendix A: Creating a Symbol File

1. Symbol Files allows a user to create a specialized symbol.
2. To create a symbol, complete the following steps.
    a. Select **File → New**
    b. Select **Block Diagram/Schematic File**
    c. Click the **OK** button.
    d. Draw your logic circuit diagram.
    e. **Save** your file with the desired name of the symbol.
    f. **Compile** your file.
    g. Select your entire file (or type CTRL+A).
    h. Select **File → Create/Update..**
    i. Select **Create Symbol File from Current File**
    j. Select **File → Close**.
    k. Now the symbol has been created.

3. To insert your symbol, complete the following steps:
    a. Select **File → New**
    b. Select **Block Diagram/Schematic File**
    c. Click the **OK** button.
    d. Double-click on the block diagram.
    e. Next to the **Name Box**, click on the **…** button. See Figure 10 below.
    f. Locate the create symbol and click **OK**.

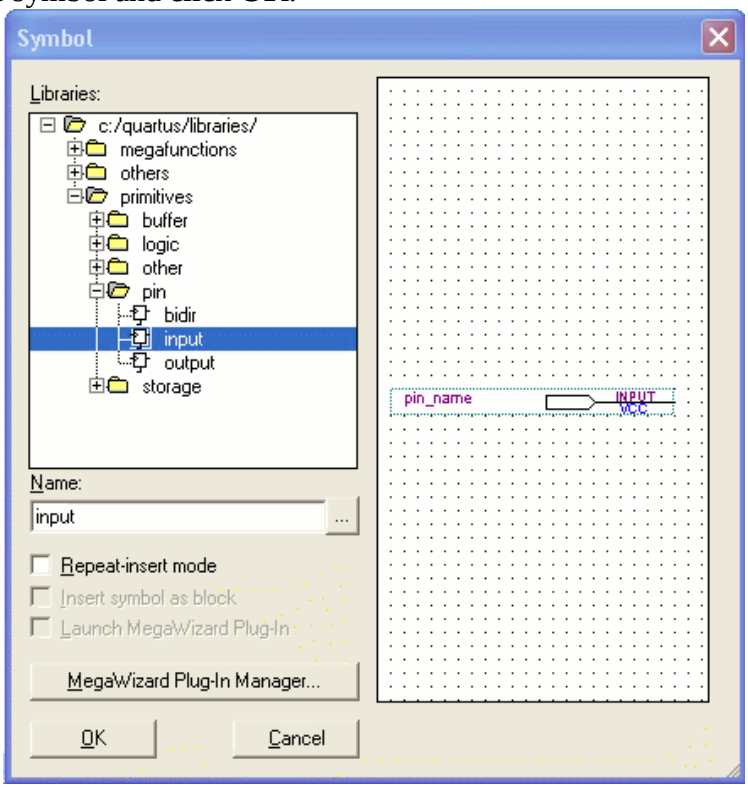
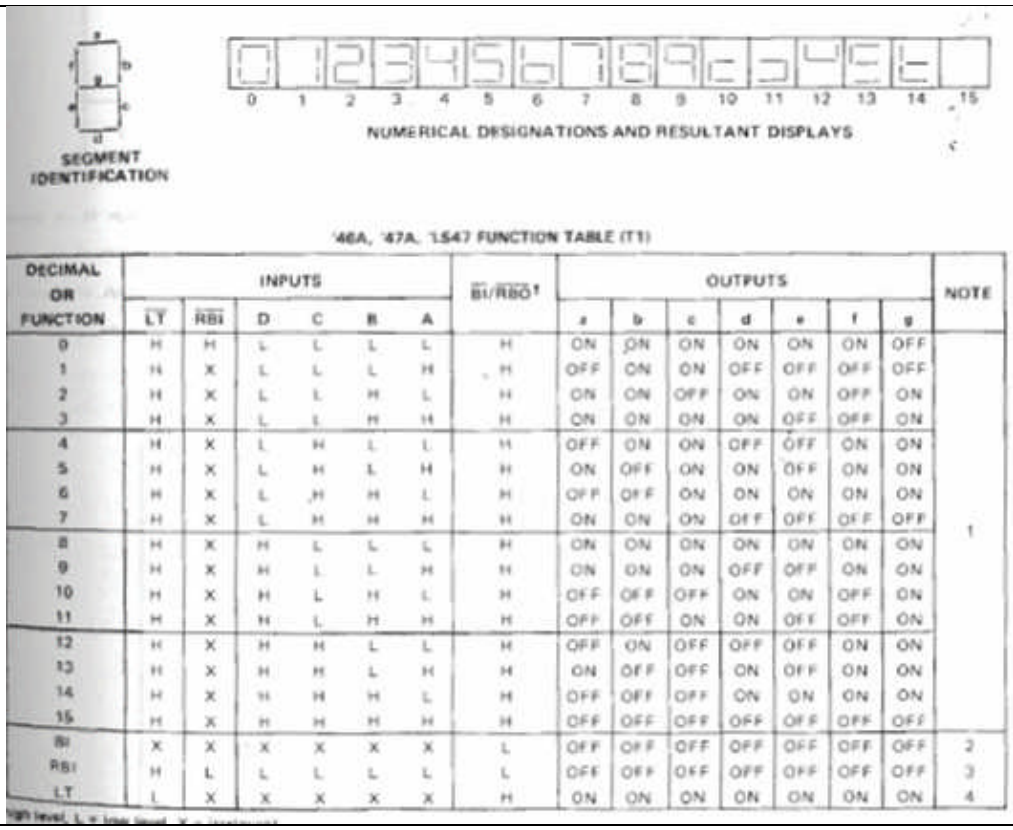


Figure 10

Appendix B: 7447 chip – BCD-to-7-segment display decoder

1. The 7447 chip is designed to convert a Binary-Coded Decimal input into the output segments of a 7-segment display.
    a. The truth table for the 7447 chip is given in Figure 11.
2. Figure 12 below gives the structure of the “7447” in Quartus II.
    a. The structure of the inputs is that “D” is the MSB and “A” is the LSB.
    b. The inputs “LTN”, “RBIN”, “BIN” must be tied to +Vcc for the 7447 chip to function properly.
    c. Each output “a” through “g” must be tied to an output block.
    d. Pin assignments must follow the system given in Table 2.



DECIMAL OR FUNCTION	INPUTS						BI/RBO†	OUTPUTS							NOTE
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	1
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

Figure 11: Truth Table for “7447” chip

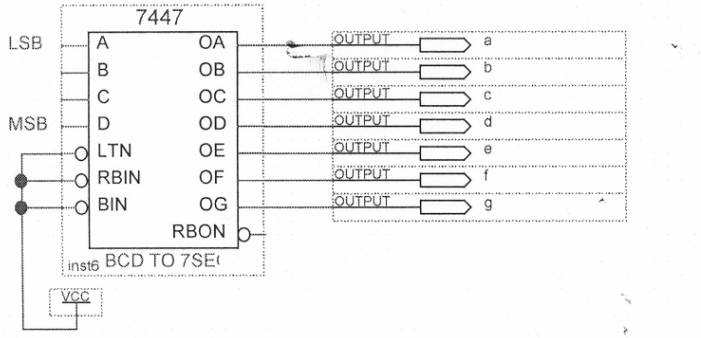
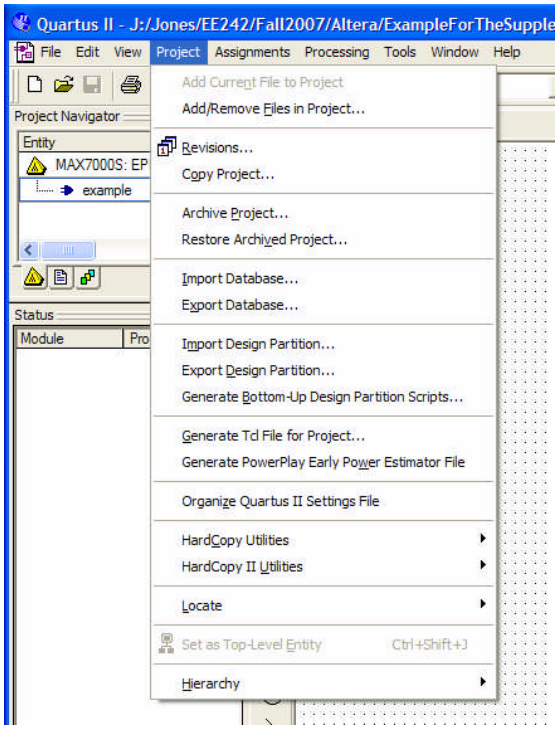


Figure 12: Structure of 7447 chip

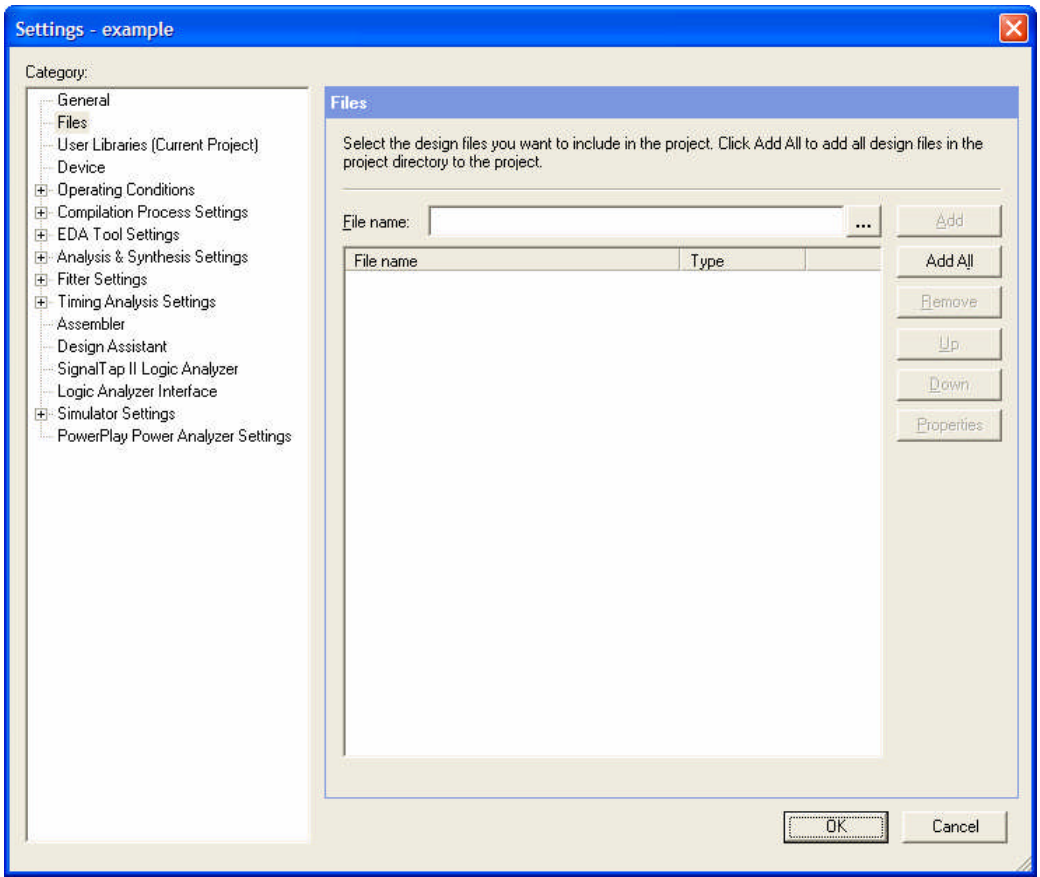
Adding Files to an Existing Project

Let's suppose you've been tasked to implement a lab that requires, among other things,a 10-bit adder. You start your project using the wizard, and you're well into the design when you realize "Wait, I've already made a FA before, so I can use this to implement a 10-bit adder." The problem is, you forgot to tell the wizard this upfront, and you're really upset!

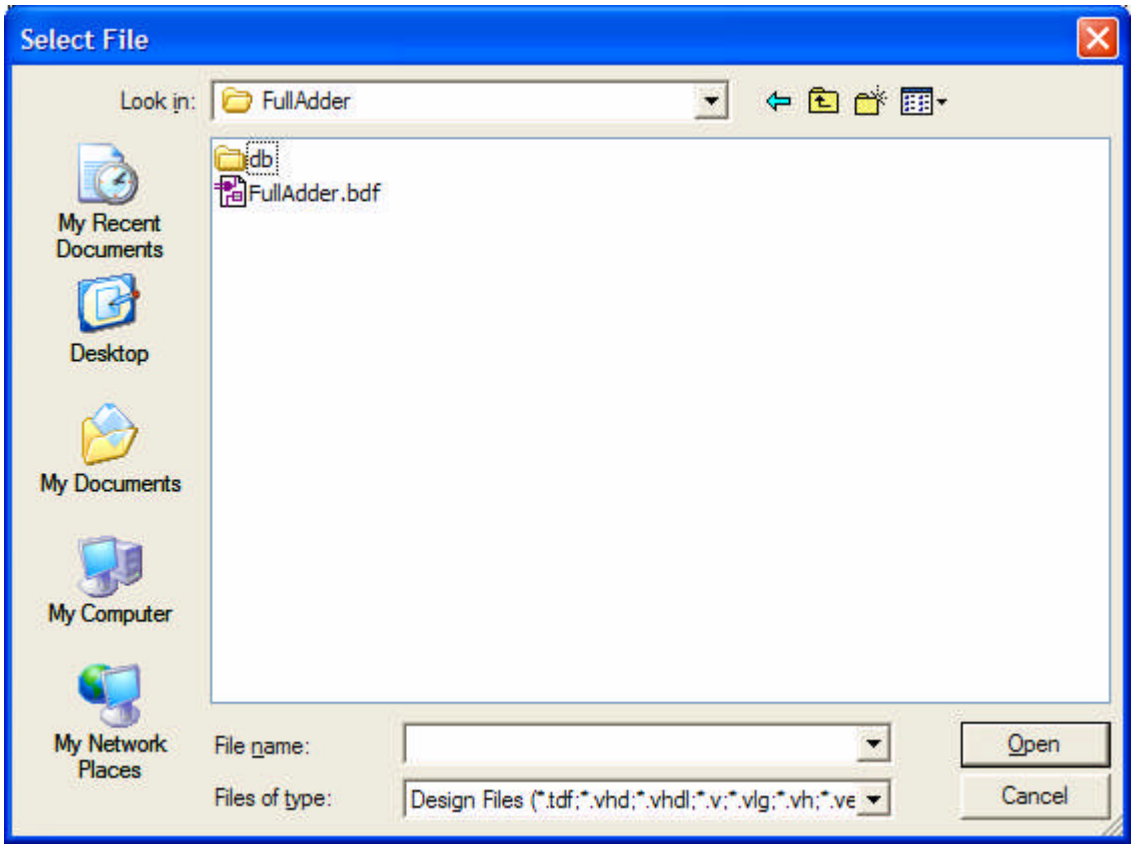
Here's how to proceed: goto the project menu, and then select the **"Add/remove Files in Project**"



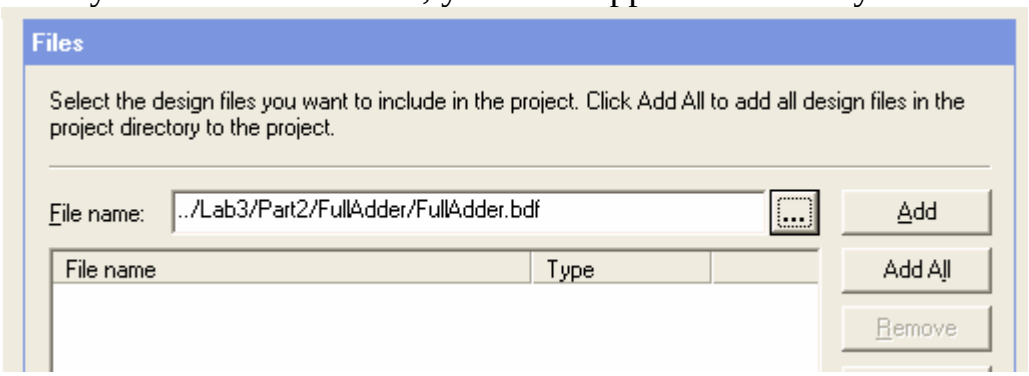
The window show below will appear. Click on the three dots "…" on the same line as the "File Name" entry box. Then navigate to the directory where your FA project is.



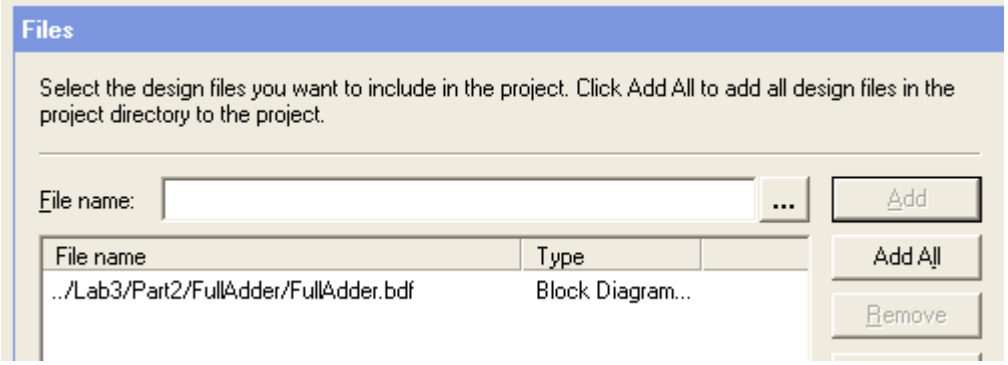
Here, you've navigated to your FA project directory, and you see the FullAdder.bdf file:



After you select the .bdf file, you see it appear in the entry box:



Then you click 'Add', and you get this:



If you have more files (from other projects for example) you can do this process again.