Design 1.2 - A Three-Bit Adder

Your next design will consist of an adder module which sums two three-bit numbers and a carry input and displays the four-bit sum on a seven-segment LED (Figure 6).

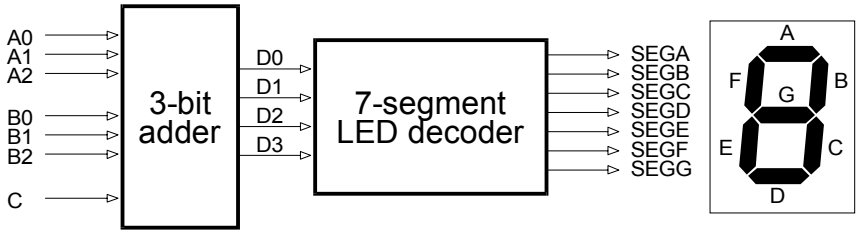
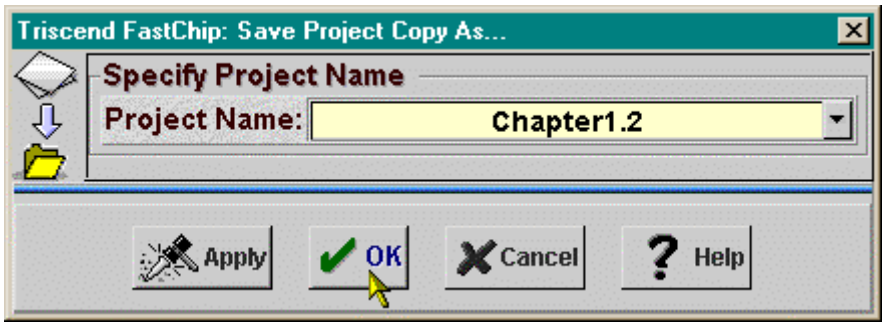


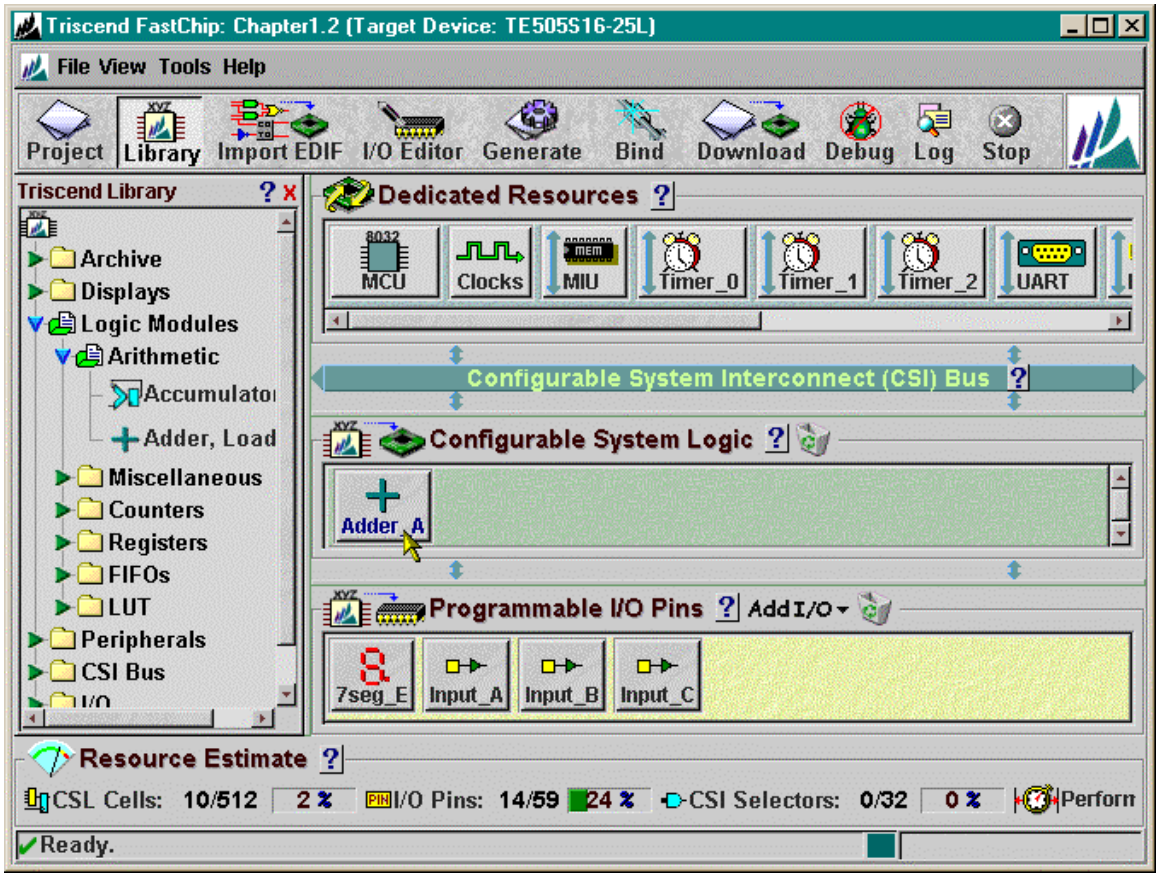
Figure 6: Block diagram of a three-bit adder that displays its sum on a seven-segment LED display.

You can use the previous design as a starting point for this design. Start FastChip and open the ***Chapter1.1*** project. Click on File⇒Save Project Copy As... and type `Chapter1.2` in the Project Name field. Then click on OK.

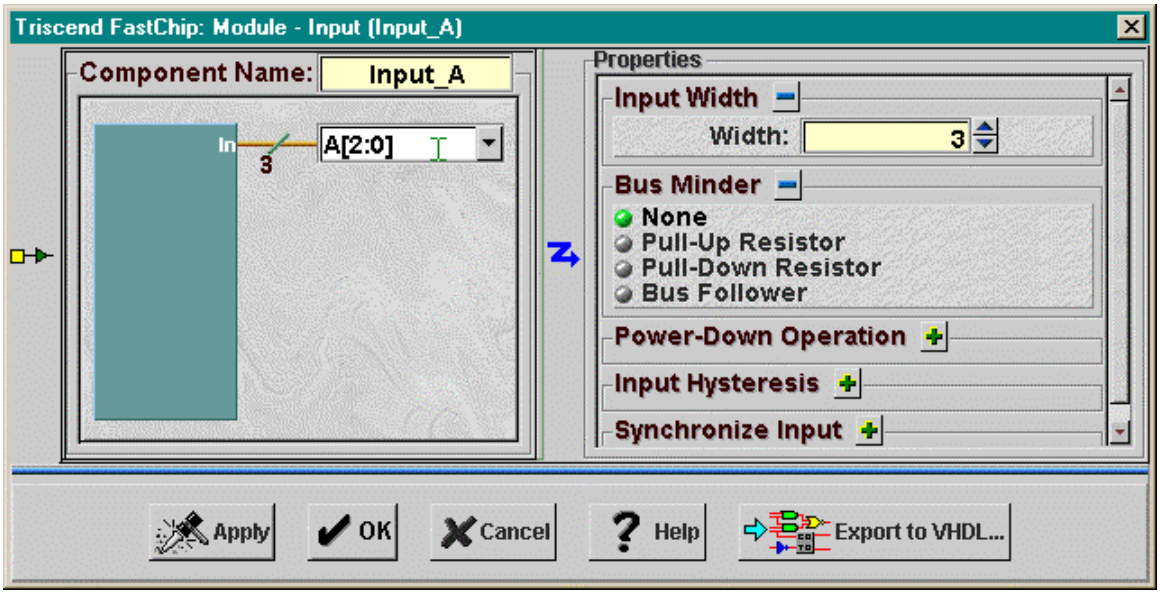


After this, your FastChip project name is changed to ***Chapter1.2***. Click on the Library icon in the toolbar to display the library of soft modules.

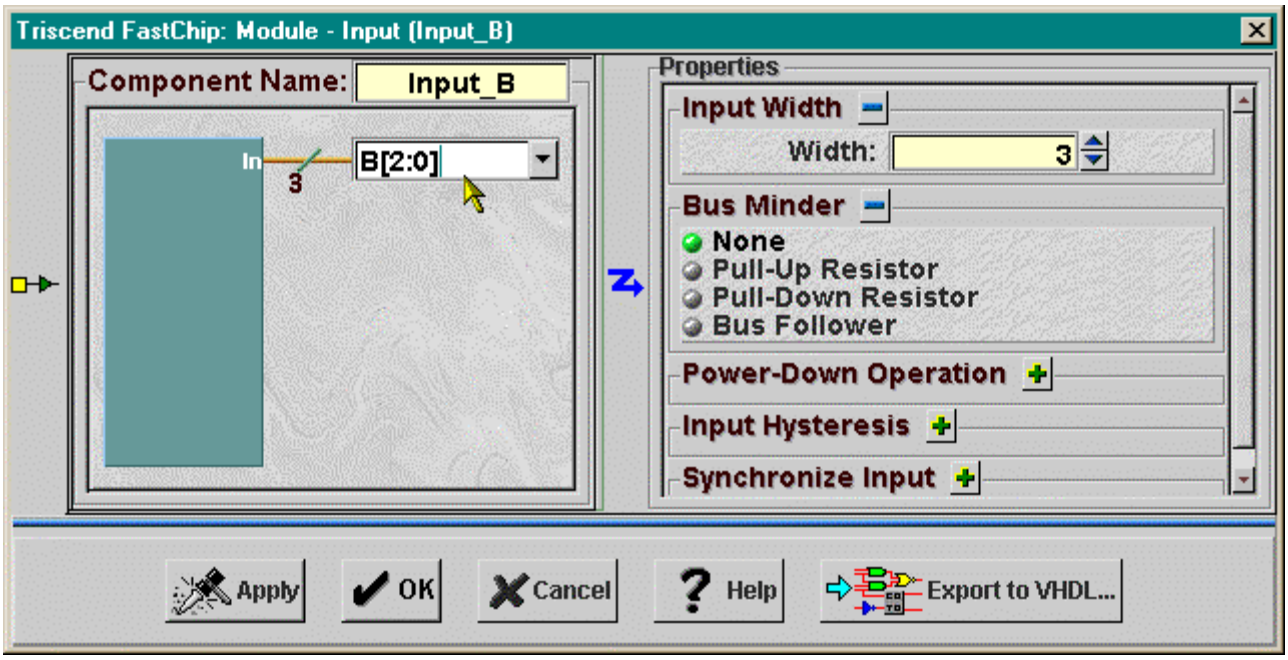
Your new design requires three sets of inputs, so click-and-drag two more Input icons from the Triscend Library pane and drop them into the Programmable I/O Pins area. Next, select an Adder icon from the Logic Modules⇒Arithmetic section of the Triscend Library pane and drop it into the Configurable System Logic area.



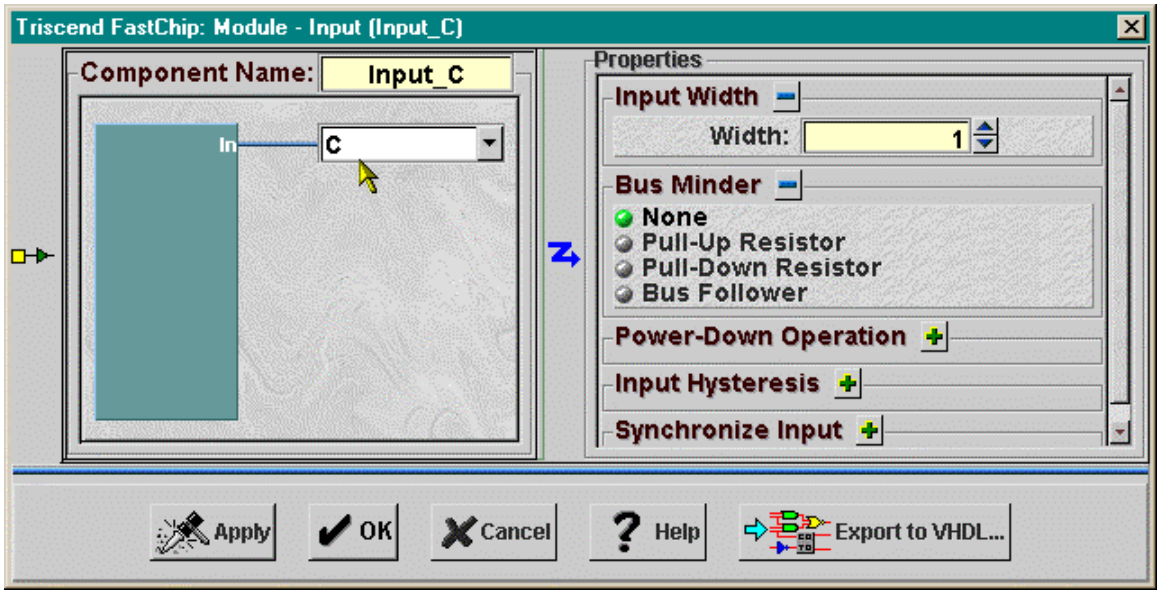
Now you need to specify the signals that interconnect the modules. Click on the Input_A icon and then decrease the bus from four to three bits by clicking on the ▼ arrow on the right-hand side of the Width field. As a result, the input signal name changes from **A[3:0]** to **A[2:0]**. This is all you need to change, so click on OK to close the window.



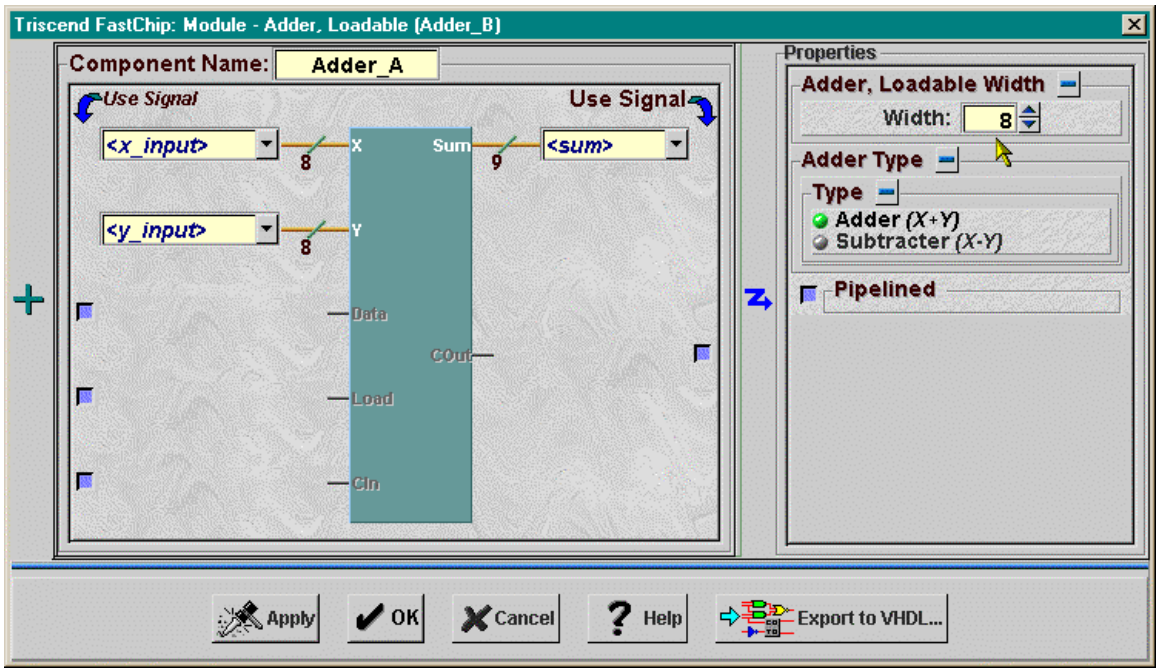
Next, click on the Input_B icon. Click twice on the ▲ arrow to increase the input width to three bits. Then type B into the <input> field as the signal name. Click with your mouse outside the field and the window should appear as follows. Then click on OK to close the window.



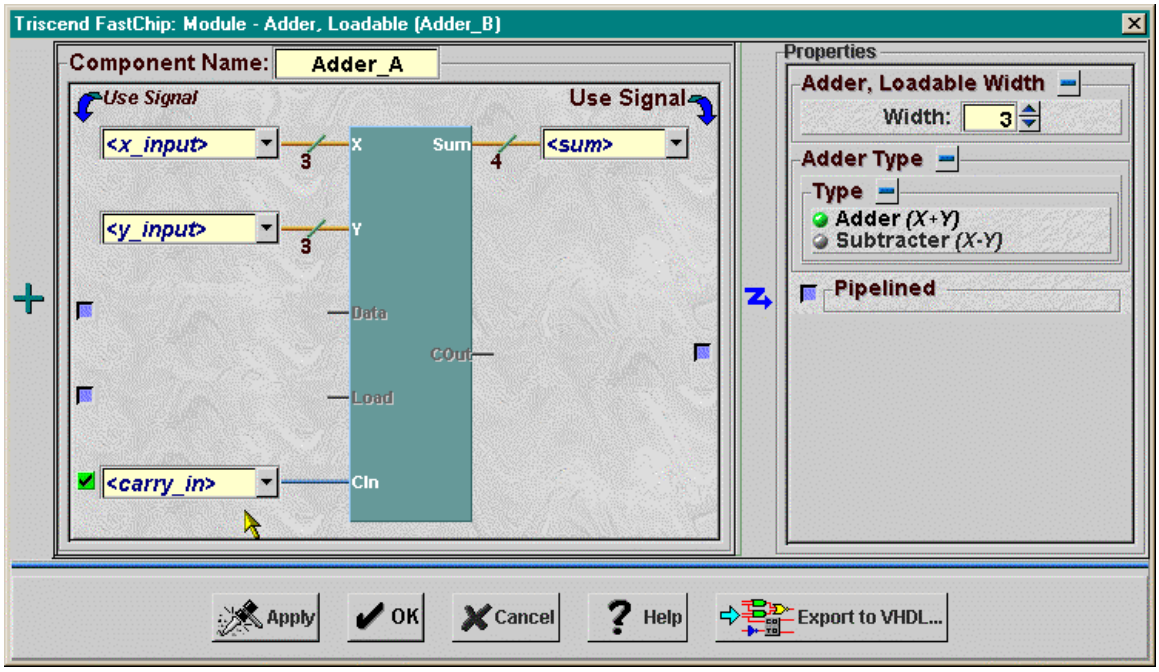
Set up the last input by clicking on the Input_C icon. This is a one-bit carry input, so there is no need to adjust the width of the signal. Just type C into the <input> field as the signal name and click on OK to close the window.



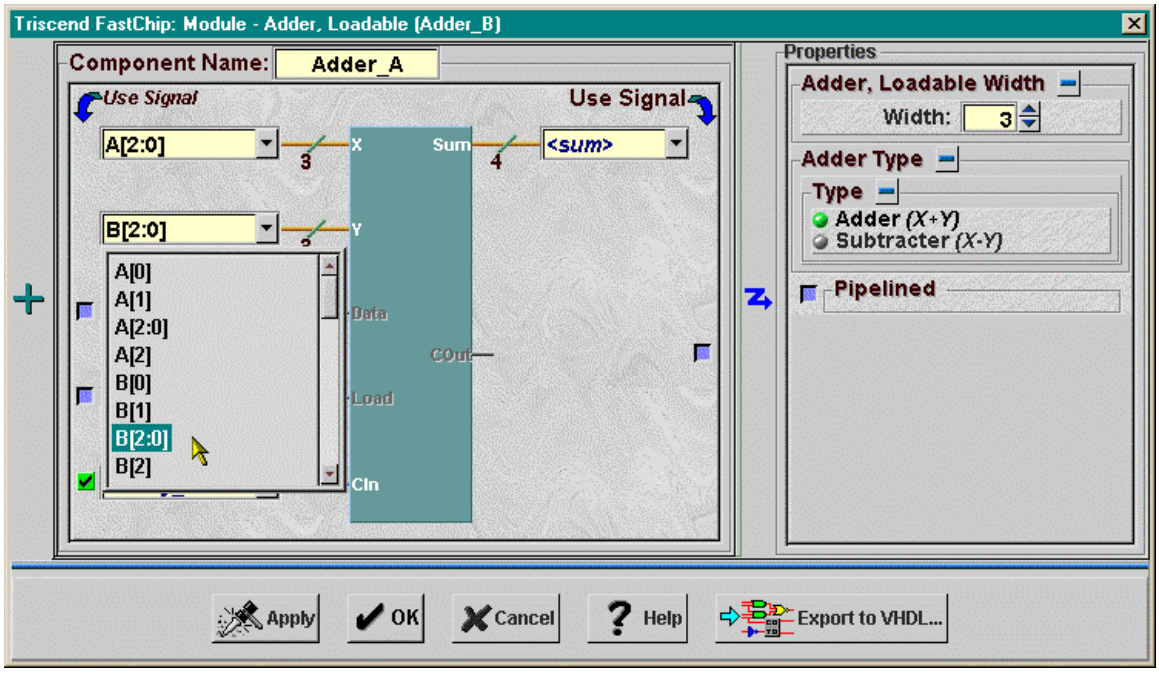
Now you can customize the **Adder_A** module for your design. Click on the Adder_A icon and the following window appears. The adder is initially set up to add two eight-bit inputs and output a nine-bit result.



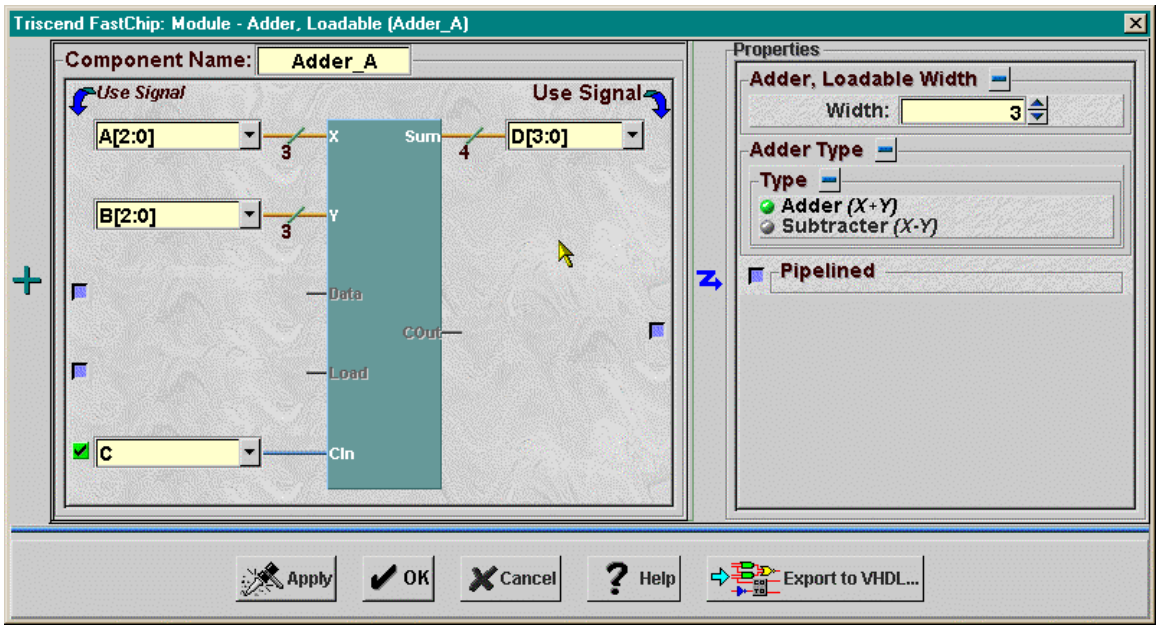
Click in the Width field and type 3 to adjust the adder to the appropriate size for this design. Then click on the checkbox to the left of Cin to enable the carry input to the **Adder_A** module.



Click on the ▼ next to the right of each signal name field and select the appropriate signal for each input and output from the list that appears.



The final assignment of signals to the inputs and outputs of the **Adder_A** module is shown below. The four-bit sum is passed to the LED decoder over the **D[3:0]** signal bus. **Adder_A** is set up so click on OK to close the window.

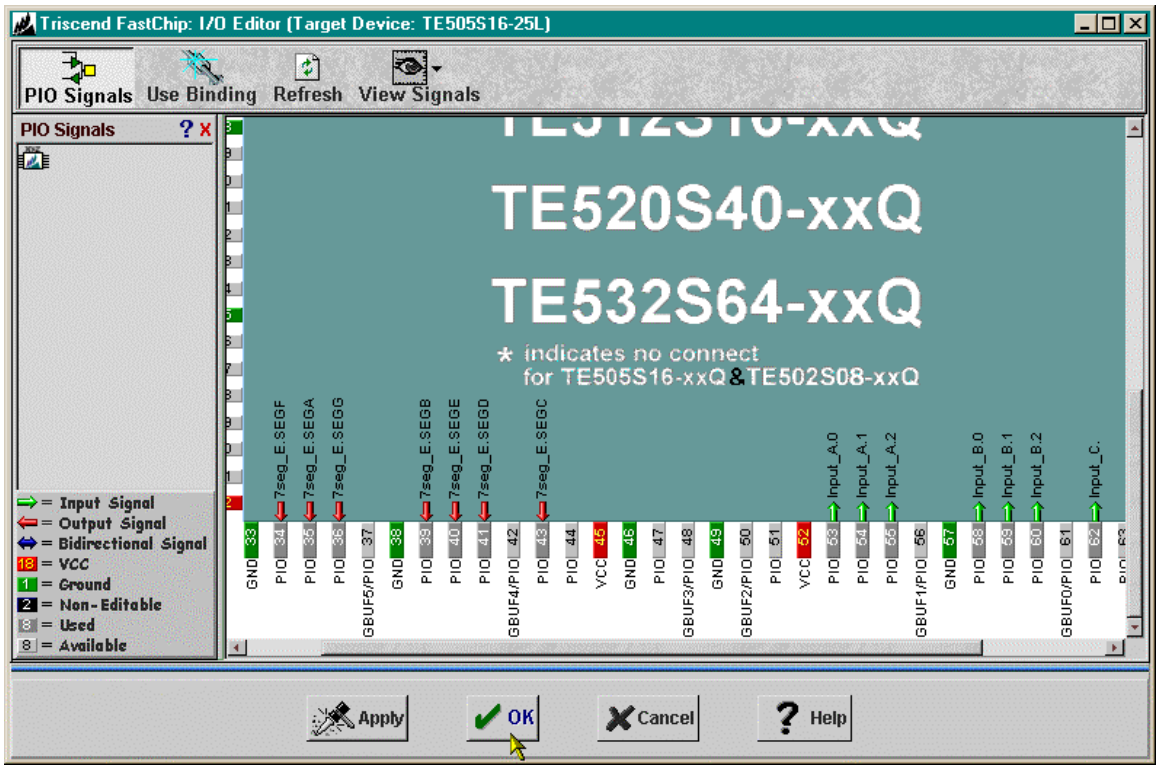


Once the modules are interconnected, the next step is to connect the I/O for your design to the pins of the CSoC. Click on the I/O Editor icon in the toolbar to display the **I/O Editor** window. The outputs for the **7seg_E** module were already assigned to the correct pins for the CSoC Board in the ***Chapter1.1*** project, and those assignments were copied over to this project when it was created. The same goes for the three **Input_A** inputs. So you only need to click-and-drag the **Input_B** and **C** signals to the pins connected to the DIP switch on the CSoC Board. The pin assignments for the inputs are as follows:

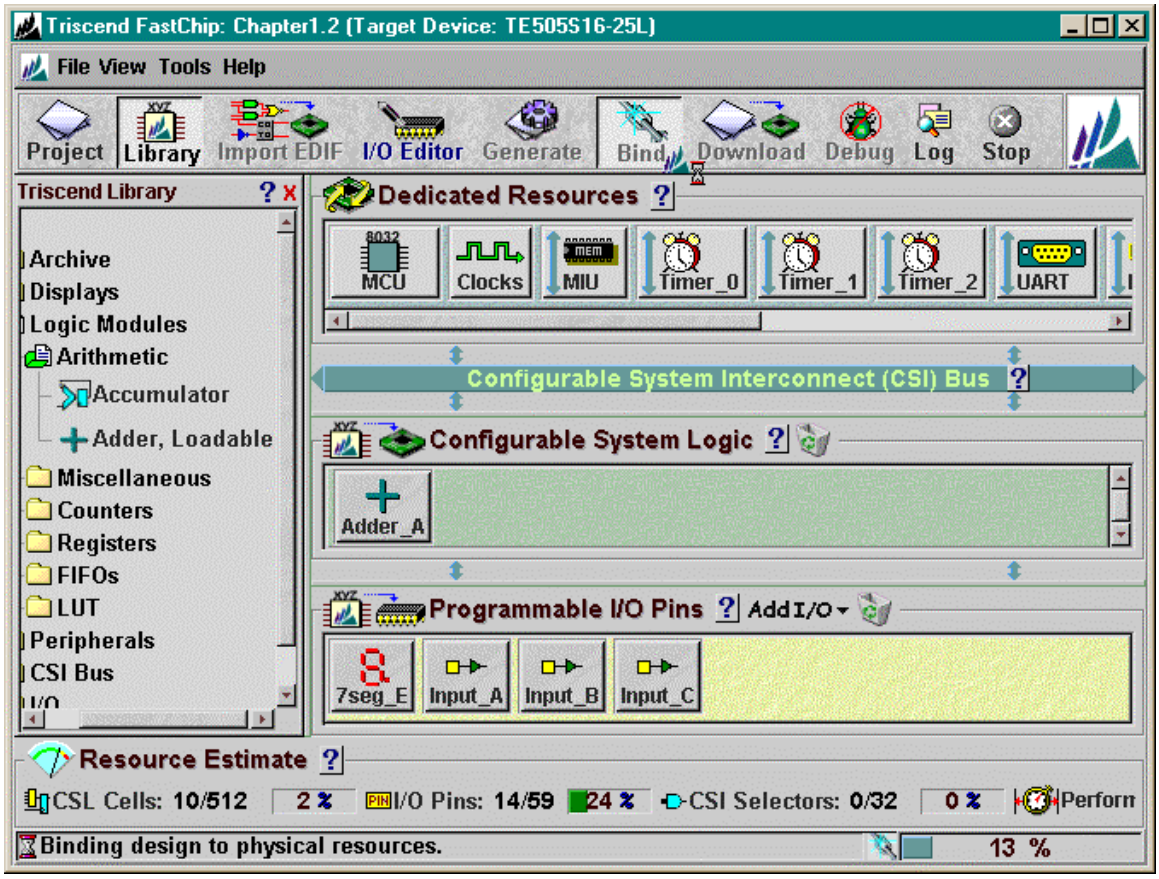
Table 2: Pin assignments for the inputs to the three-bit adder design.

Signal	Pin	CSoC Board Resource
Input_A.0	53	DIP switch position #1
Input_A.1	54	DIP switch position #2
Input_A.2	55	DIP switch position #3
Input_B.0	58	DIP switch position #4
Input_B.1	59	DIP switch position #5
Input_B.2	60	DIP switch position #6
Input_C	62	DIP switch position #7
not used	63	DIP switch position #8

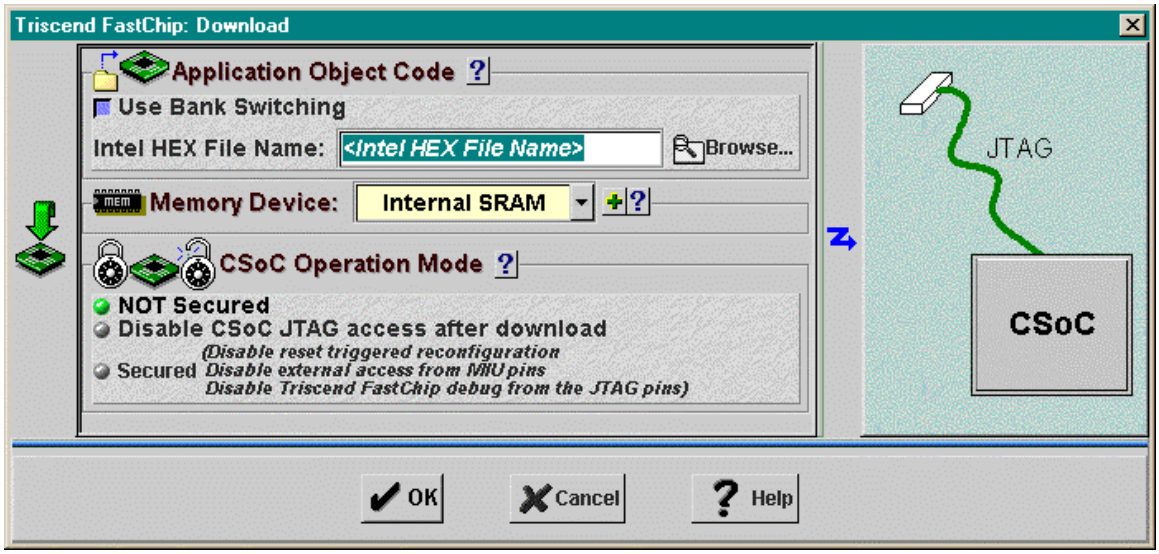
The **I/O Editor** window looks like this after the pin assignments are made:



Once the pin assignments are complete, you can map, place, and route your circuit to the CSL of the CSoC by clicking on the Bind icon on the toolbar. Note from the status bar that this design consumes ten cells in the CSL (seven for the LED decoder and three for the three-bit adder) and fourteen of the I/O pins (seven for the LED decoder outputs, six for the two 3-bit addends and one for the carry input).



You can download your design once the bind process completes. *As always, make sure your CSoC Board is attached to the 9V DC power supply and it is connected to the parallel port of your PC with the downloading cable.* Click on the Download icon on the toolbar and then click on OK in the **Download** window that appears. This initiates the downloading of the configuration file into your CSoC Board.



With your design downloaded into the CSoC Board, you can try some test cases to see if various inputs are summed correctly. The first addend is driven by the first three DIP switches, the second addend is driven by the next set of three switches, and the carry input is set by the seventh switch. Figure 7 shows a few test examples.

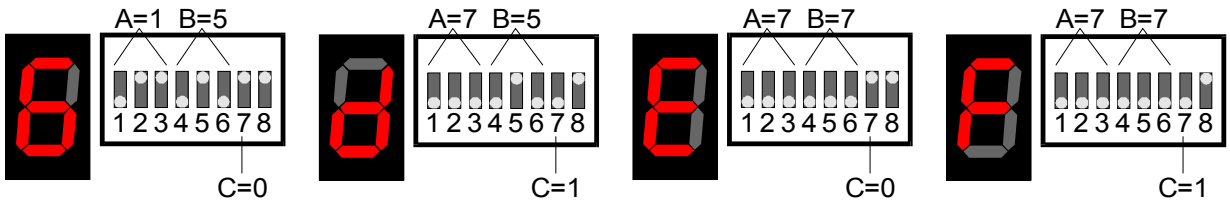


Figure 7: Testing the three-bit adder.

Once you are done testing the adder, finish by clicking on File⇒Save Project and File⇒Exit.

Design 1.3: A Simple Timer

Your previous designs were combinatorial in nature: the output was a function of only the current inputs. This design introduces a sequential component, a counter, whose output depends upon its previous state.

The counter is used to build a timer where the elapsed time is displayed on the seven-segment LED (Figure 8). A 25 MHz clock drives a 27-bit counter. The counter will roll-